Journal of King Saud University – Computer and Information Sciences 34 (2022) 1869–1883

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A new attributed graph clustering by using label propagation in complex networks

Kamal Berahmand [a,*], Sogol Haghani [b], Mehrdad Rostami [c], Yuefeng Li [a]

[a] Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia
[b] Department of Computer Engineering and Data Mining Laboratory, Alzahra University, Vanak Tehran, Iran
[c] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Article history:
Received 13 May 2020
Revised 31 July 2020
Accepted 18 August 2020
Available online 1 September 2020

Keywords:
Complex network
Attributed graph clustering
Label propagation
Node similarity

ABSTRACT

The diffusion method is one of the main methods of community detection in complex networks. In this method, the use of the concept that diffusion within the nodes that are members of a community is faster than the diffusion of nodes that are not in the same community. In this way, the dense subgraph will detect the graph in the middle layer. The LPA algorithm, which mimics epidemic contagion by spreading labels, has attracted much attention in recent years as one of the most efficient algorithms in the subcategory of diffusion methods. This algorithm is one of the detection algorithms of most popular communities in recent years because of possessing some advantages including linear time order, the use of local information, and non-dependence on any parameter; however, due to the random behavior in LPA, there are some problems such as unstable and low quality resulting from larger monster communities. This algorithm is easily adaptable to attributed network. In this paper, it is supposed to propose a new version of the LPA algorithm for attributed graphs so that the detected communities solve the problems related to unstable and low quality in addition to possessing structural cohesiveness and attribute homogeneity. For this purpose, a weighted graph of the combination of node attributes and topological structure is produced from an attributed graph for nodes which have edges with each other. Also, the centrality of each node will be calculated equal to the influence of each node using Laplacian centrality, and the steps of selecting the node are being enhanced for updating as well as the mechanism of updating based on the influence of nodes. The proposed method has been compared to other primary and new attributed graph clustering algorithms for real and artificial datasets. In accordance with the results of the experiments on the proposed algorithm without parameter adjusting for different networks of density and entropy criteria, the normalized mutual information indicates that the proposed method is more efficient and precise than other state-of-the-art attributed graph clustering methods.

1. Introduction

Complex networks are essential tools for investigating myriad natural phenomena which are happening at present such as biological networks, brain network as well as those which have been created by human in the era of technology, including social network and network traffic that can be described by complex networks. This issue has made network science a hot, widespread, and interdisciplinary field in the present era. At the heart of these complex networks, there are many problems, such as community detection (Mohammadi et al., 2019), identifying spreader nodes (Berahmand et al., 2018a, 2019), maximal influence (Berahmand

* Corresponding author.
E-mail addresses: kamal.berahmand@hdr.qut.edu.au (K. Berahmand), s.haghani@student.alzahra.ac.ir (S. Haghani), M.rostami@eng.uok.ac.ir (M. Rostami), y2.li@qut.edu.au (Y. Li).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.08.013
1319-1578/

et al., 2018c), and link prediction (Haghani and Keyvanpour, 2019), which are considered as the main challenges. The community structure is one of the most popular and vital topological properties of complex networks. The community detection is partitioned into several densely connected sub-graphs that facilitate to understand and visualize large graphs. In recent years, with the proliferation of rich information available for real-world objects, vertices in graphs are often associated with several attributes that describe the characteristics and properties of the vertices (Gibson and Faith, 2011, Berahmand et al., 2018b). In a PPI network, an attribute value can represent gene expression data, which encodes the differential expression value of each gene when exposed to stimuli (Rostami et al., 2020). In a social network, an attribute might correspond to the personal profile of a member such as age, interests, locale, etc. (Xu et al., 2012, Greene and Cunningham, 2013). The web graph consists of web pages interweaved by hyperlinks. Each web page is also characterized by a series of attributes, including URL, name, keywords, contents, tags, and other six items. This issue leads to a new type of graph, called attributed graphs, and hence the demand for a new clustering task named attributed graph clustering. The topological network structure shows the interactions between vertices, and the node attribute information represents the common characteristics among nodes. They both play essential roles in the formation of a network community structure. However, nowadays, most community detection algorithms only use the topological network structure. The community detection in such attributed networks using both network topological structure and node attribute information is crucial yet challenging and relies strongly on appropriate similarity learning (Huang et al., 2015, Fang et al., 2016). An ideal attribute graph clustering should generate clusters that have dense subgraphs with cohesive intra-cluster structure and homogeneous vertex properties by balancing the structural and attribute similarities (Li et al., 2017).

Although the graph clustering has been investigated extensively, the clustering analysis of large graphs with rich attributes remains a major challenge in practice (Yang et al., 2009). An attributed graph clustering is composed of a group of nodes that the similarity among nodes is maximal. In this regard, nodes' similarity can be computed based on two measures, including structural similarity and attribute similarity (Amiri et al., 2018). The structural similarity is extracted based on network topology. The attribute similarity is computed using the individual nodes' internal characteristics that are entirely independent of the network topology. The topological network structure reflects the interactions between nodes, and the node attribute information reflects the common characteristics among nodes. They both play essential roles in the formation of the network community structure (Karimi-Majd and Fathian, 2017, Alinezhad et al., 2020). As mentioned earlier, the structural similarity measure is extracted using network topology that can be computed by different approaches such as k-distance neighborhood and common neighbors. In k-distance neighborhood approach, two nodes are similar, if the distance (i.e., Manhattan distance) between them is less than k. In common neighbors approach, two nodes are similar, if they have more common neighbors, even if they are not adjacent. As noted earlier, the attribute similarity considers the internal characteristics of nodes without any knowledge about network topology and graph structure. For instance, a person node's internal attributes in social networks can include the date of birth, sex, affiliation, age, and occupation (Zarandi and Rafsanjani, 2018). The authors argue that the useful clustering analysis of a large graph with rich attributes requires a systematic graph clustering analysis framework that partition the graph based on both structural similarity and attribute similarity.

The problem of community detection is generally divided into two categories: only using structural information and the combination of structural information and attributes. In the first category, the network structure is traditionally employed to identify the dense subgraph; some popular methods include modularity (Clauset et al., 2004), Label Propagation Algorithm)LPA) (Raghavan et al., 2007), and random walk (Pons and Latapy, 2005, Rosvall and Bergstrom, 2008). However, in contrast to these methods, some methods use structural information and network attributes, and the structure-attribute combined graph clustering method aims at partitioning a graph with rich attributes into several clusters with cohesive intra-cluster structures and homogeneous attribute values; SA-cluster (Zhou et al., 2009), CODICIL (Ruan et al., 2013), and PCL-DC (Yang et al., 2009) methods are among popular methods in this category. LPA method is one of the most popular community detection methods in recent years, which has been based on the structural network. This method has been popular because of its low time complexity (linear order) and low spatial complexity (using local information). This algorithm has problems such as unstable and lower quality in community detection due to random behavior, and many algorithms have been proposed to enhance its random mode. The present paper, intends to propose the Structure-Attribute Similarities Label Propagation (SAS-LP) algorithm, which is the adaptable version of the LPA algorithm for attributed networks. Also, this algorithm's problem resulting from random behavior will be solved by employing the capabilities of attributed networks.

The main contributions of this work are summarized in the following five folds:

- SAS-LP is an adaptable algorithm for attributed networks and reduces the iteration times and keeps the original time efficiency.
- The SAS-LP algorithm has better stability than the LPA algorithm. In this manner, the stability of the algorithm is ensured. Simultaneously, the nodes with small influence back disturbing the nodes with significant influence and unnecessary iterative updates are avoided.
- A new similarity measure for each pair of vertices has been proposed in attributed networks that consider both the graph structure and attribute information.
- A new measure of node influence has been proposed in attributed networks, which can play the role of clusters center.
- The experimental results of accuracy and efficiency on the synthetic benchmarks and the real-world networks show that the algorithm's performance is significantly better than each comparison algorithm, which is suitable for the large-scale complex network.

The rest of the paper is organized as follows. Section 2 summarizes related works to attributed graph clustering in a complex network. In Section 3, some preliminaries of this work are introduced, such as the definition of attributed and structural similarity, label influence, label acceptance, and the proposed algorithm (SAS-LP) in detail. The results of simulation and experimental analysis are explained in Section 4, and the conclusion is given in Section 5.

2. Related work

The community detection has been comprehensively studied in the literature. In this section, the existing popular community detection algorithms are summarily introduced. An excellent sur-

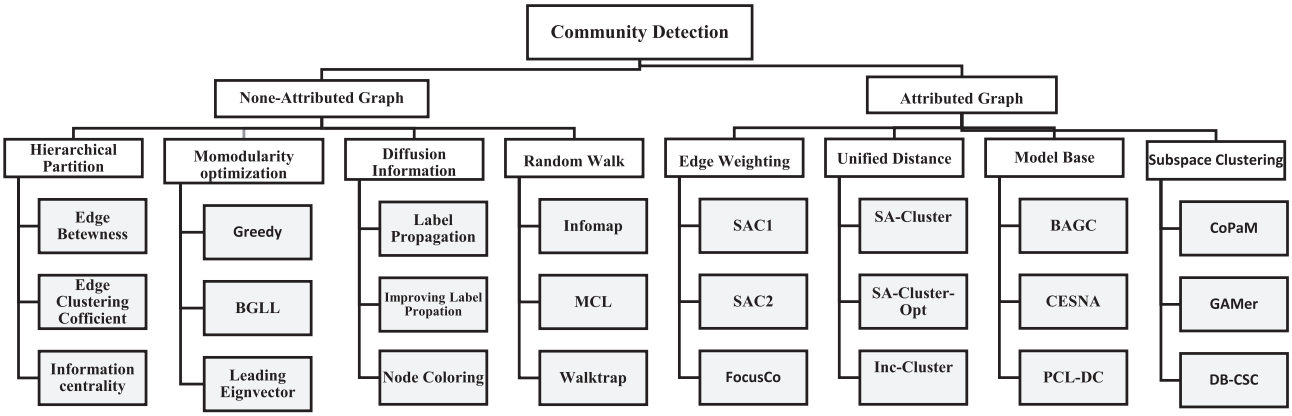

Fig. 1. Category of structural clustering and attributed clustering.

vey is available in (Bothorel et al., 2015, Chunaev, 2019). The authors categorize existing community detection algorithms in two groups:1) Non-Attributed graph mainly focused on the connectivity structures based and ignored attributes of the nodes, 2) Attributed graph dealing with both structure and attribute information. The Non-attributed graph divided into four main groups including, a) Hierarchical clustering, b) Modularity-based methods, c) Random walk based approach, and d) Label propagation-based algorithms, and attributed graph is divided into four well-known groups including a) Edge Weighting, b) Augmented Graph, c) Quality Function Optimization, d) Unified Distance. This category of community detection including two main branches is indicated in Fig. 1.

Hierarchical clustering is a popular and oldest method for finding communities (clusters) in complex network analysis. The starting point of any hierarchical clustering method is the definition of a similarity measure. Hierarchical clustering methods can be classified into two categories, including agglomerative algorithms and divisive algorithms for finding communities. In random walk methods, each node contains a walker initially. Then each walker will randomly choose a neighbor of the node it currently stands on to localize. The idea behind the random walk is that the walk tends to be trapped in dense parts of a network to communities (Pons and Latapy, 2005). Currently, many random walk base community detection methods have been proposed, such as MCL (vanDongen, 2000), Walktrap (Pons and Latapy, 2005), and Infomap (Rosvall and Bergstrom, 2008) algorithms. Modularity-based methods try to detect communities based on modularity metric. These methods suppose a high modularity value for a well-separated community. It is evident that the number of methods to partition n nodes into k non-empty groups is given by the Sterling number of the second kind (k); hence, the number of distinct community divisions is the Bell number. Therefore, it is proven that modularity optimization is an NP-complete problem. The purpose of all modularity-based methods is to discover a partition of the network so that the modularity value is maximized. The proposed methods of modularity maximization can be classified into three main categories: greedy-based, heuristic methods, and spectral optimization. Label propagation algorithm (LPA) is a popular and fast method for community detection proposed by Raghavan et al. (2007). Initially, a unique label is assigned for each node in the network. In the next step, each node updates its label with the most frequent label present among its neighbors. When some labels of neighbors are equally frequent, the algorithm randomly selects among the most frequent labels. This label propagation process is repeated until the nodes with the same label are grouped into one community. LPA's main advantages including possessing a nearly-linear time complexity, using the local information, independency from free parameters and the objective function, and simplicity of implementation (Sun et al., 2015). However, this algorithm has shortcomings, such as instability, low quality, and forming monster communities due to its random behavior in initial node selection and randomly updating the label of a node in the tie break state. Recent investigations reveal that numerous modified versions of LPA have been devised to improve its stability and robustness (Berahmand and Bouyer, 2018, Zhang et al., 2018, Berahmand and Bouyer, 2019, Garza and Schaeffer, 2019). The above-mentioned graph clustering and summarization approaches consider only one aspect of the graph properties but ignore the others.

Attributed graph clustering utilizes information from both structures and attributes to find clusters in graphs. These clusters are groups of densely connected nodes and are highly similar in their attributes as well. Many graph clustering approaches have been proposed to utilize content information besides structure information of graphs. The authors divide these approaches into four categories: (1) approaches that convert an attribute graph to a weighted graph, (2) distance-based approaches, (3) model-based approaches, and (4) subspace-based approaches. Finally, a summary of selected approaches in each group will be presented. The first category includes approaches based on a conversion of the originally attributed graph to a weighted graph, such as FocusCo (Perozzi et al., 2014). Node attributes are removed from the nodes by storing their information inside the edges of the graph, which is performed by giving an attribute a similarity value between two nodes in the edge of nodes as the weight of it. The second category includes distance-based approaches such as the SI-Cluster (Zhou and Liu, 2013), SA-Clustering (Zhou et al., 2009), and CODICIL (Ruan et al., 2013). The structure information is stored in a similarity (distance) function between nodes, and it is combined with the attribute similarity (distance) function. The third category is related to model-based approaches, which e include, but are not limited to, PCL-DC (Yang et al., 2009), Bayesian probabilistic model (Xu et al., 2012) and CESNA (Yang et al., 2013). They are based on a probabilistic model that avoids the artificial design of a distance measure. The fourth category is the subspace clustering approaches. A selected set of proposed approaches in this category includes CoPaM (Moser et al., 2009), GAMer (Gunnemann et al., 2010), DB-CSC (Günnemann et al., 2011) and SSCG (Günnemann et al., 2013). They identify the clusters only on the

context of their relevant features as a subset of all nodes' attributes, especially for high dimensional data.

2.1. The approaches which convert an attribute graph to a weighted graph

Attribute similarity between nodes of a graph may show the strength of the relationship. Hence, node attributes are removed from the nodes by storing their information inside the edges of the graph. This is performed by giving attribute similarity between two nodes in the edge between them as the weight of it. After reshaping the graph as a weighted graph, different graph clustering algorithms, considering the weight of edges, can be applied to this weighted graph. If the edges are maintained with high weights during the clustering process, groups of nodes with similar attribute values can be obtained.

2.2. Distance-based approaches

The simplest idea for attributed graph clustering is to define some vertex-wise distance metric that considers both the structure and attribute information of vertices in a graph. For instance, the differences in vertex attribute values can be quantified as distances between neighboring vertices (Steinhaeuser and Chawla, 2010). The textual web content and hyperlinks are also combined in a similarity measure for web page clustering. Different similarity (distance) measures are proposed for this purpose, and classic distance-based clustering methods can be applied to the graph data using these proposed measures.

2.3. Model-based approaches

Another stream of related work for attributed graph clustering builds primarily upon generative probabilistic models, the structure and vertex attribute information are correlated to a set of shared, and hidden variables of cluster membership within each graph (Yang et al., 2013). This approach avoids the artificial design of a distance measure.

2.4. Subspace clustering

More recent methods use unsupervised feature selection as the subspace clustering and extract cohesive subgraphs with homogeneity in a subset of attributes. Subspace clustering methods are proposed to solve this problem by identifying clusters only in terms of their relevant features, especially for high dimensional data. However, finding relevant features is computationally difficult. An optimization step is required to combine different quality measures such as density, entropy, and dimensionality. A summary of some selected approaches in this category is presented below, which combines the subspace clustering with dense subgraph mining.

3. Proposed method

Before addressing the algorithm, let us review some definitions and concepts, which are the foundations of the proposed algorithm.

3.1. Background and notation

Generally, an attributed network can be represented by the triple $G = (V, E, A)$, where V is the set of nodes, E the set of edges representing the existing relations between the nodes, and A implies the set of attribute vectors. The value of $n=|V|$ is the total number of vertices, $m=|E|$ is the total number of edges, and A ($attr_1$, $attr_2$, $attr_3 \ldots$ $attr_k$) associates with nodes in V and describes their features. K is the dimension of attribute vectors of A. In this the present work, the authors focus on graphs with binary (interchangeably, label) attributes on nodes. The structural similarity is extracted based on a network topology that is the most important similarity measure in community detection.

Definition 1 (*Attribute Similarity (ASIM)*). The attribute similarity is computed using the individual nodes' internal characteristics that are entirely independent of the network topology. Here the Simple Matching Coefficient criterion (Faizal, 2014) is used to calculate the similarity of attributes. The Simple Matching Coefficient (SMC), is a statistical measure for correlating the similarity between binary data samples. The matching coefficient is only applicable to the graph with the categorical node attribute. It is defined as the ratio of the total number of matching attributes to the total number of present attributes, calculated by Eq. (1) as follows:

$$\mathrm{SMC}(\mathrm{i},\mathrm{j}) = \frac{\text{number of matching attribute}(\mathrm{i},\mathrm{j})}{\text{total number of attributes}(\mathrm{i},\mathrm{j})} \tag{1}$$

In which are node (i) and node (j) described using n binary attributes.

Definition 2 (*Structural Similarity (SSIM)*). The structural similarity is extracted based on a network topology that is the most important similarity measure in community detection. The structural similarity of nodes a and b used Jaccard similarity. It is the ratio of common neighbors of nodes a and b to all neighbor nodes of a and b. As a result, the value of the Jaccard index prevents higher degree nodes from having a high similarity index with other nodes. Jaccard similarity (a,b) is computed by as Eq. (2) as follows:

$$\mathrm{Jaccard}(\mathrm{i},\mathrm{j}) = \frac{[\Gamma(a) \cap \Gamma(b)]}{[\Gamma(a) \cup \Gamma(b)]} \tag{2}$$

$\Gamma(a)$ shows the first-order neighborhood of node $a \in V$.

Definition 3 (*Weight Matrix*). Consider the attributed graph G = (V, E, A), which is converted to a weighted graph G = (V, E, W) using structural similarities and the attribute between two nodes that have edges with each other. The weight of each edge will be obtained by Eq. (3), which is calculated by the combination of the matrix of attribute similarity from Eq. (1) and matrix of structural similarity from Eq. (2). The weight of the edge between two nodes (i, j) is a measure that quantifies the closeness between nodes. The combination of these two types of similarities is used to increase the accuracy of the weight of the link between two nodes. The more accurate the weight of an edge, the detection of nodes with more considerable influence will be easier in terms of structure and attributes by employing the Laplacian centrality (Qi et al., 2012). Structural and non-structural similarities are two completely independent issues, each with a distinct role in determining the similarity. The main challenge in these types of algorithms is the effective combination of these two types of structural and non-structural similarities to enhance the results, in which the element (i,j) will be equal to Eq. (3).

$$\begin{aligned}\mathrm{Weight}(i,j) &= \{\alpha * SSIM(i,j) + (1-\alpha) * ASIM(i,j)\} \;\; ifA(i,j) \\ &= 10, \;\; \text{otherwise}\end{aligned} \tag{3}$$

where $\alpha \in [0, 1]$ is the fusion coefficient, a hyper-parameter that influences the balance between structural and attribute components. Also, SSIM (i,j) and, ASIM (i,j) are referred to as the Structural Similarity and Attribute Similarity, respectively.

Definition 4 (*Label Influence (LI)*). Many centrality measures can identify the influence of nodes. However, most of these centralities use the structural information of the graph to determine the nodes with high influence; but, none of these centralities are proper options for attributed graphs because they ignore the information of attributes. Since the considered graph matrix is weighted using structural and attributed information, the authors use the Laplacian centrality criterion, which has been employed in recent years to calculate the centrality of nodes in weighted networks. Laplacian centrality, with its linear complexity and the use of semi-local information, is highly popular in detecting the influence nodes in weighted networks (Qi et al., 2012). In this measure, not only the direct adjacent of a node but also the importance and influence of the adjacent are considered. In order to detect the nodes, the cluster heads in the community are considerably precise. Laplacian centrality is defined as the drop of the Laplacian energy of the network with the elimination of the target node from the network. To calculate the center of Laplacein, the author first converts the graph G=(V, E, A) to a weight graph G=(N, E, W) by using Formula 3, Afterward the sum of weights of the link between each node to its neighbor is calculated to get a diagonal matrix Y(G):

$$\mathrm{Y(G)} = \begin{pmatrix} x_1 & 0 & 0 & \cdots & 0 \\ 0 & x_2 & 0 & \cdots & 0 \\ 0 & 0 & x_3 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & x_n \end{pmatrix} \tag{4}$$

where $x_i = \sum_{j=1}^{n} w_{ij}$ is the sum of weights of the links among nodes i and the other nodes in the network.

The Laplacein energy for the network is also calculated as Eq. (5)

$$E_l(G) = \sum_{i=1}^{n} X_i + 2\sum_{i<j} w_{ij}^2 \tag{5}$$

The Laplacian centrality $\mathrm{LAPC_i}$ of the node i can be expressed as

$$\mathrm{LAPC}_i = \frac{(\Delta E)_i}{E_L(G)} = \frac{E_L(G) - E_L(G_i)}{E_L(G)} \tag{6}$$

In this equation, $EL(G_i)$ is the Laplacian energy value of the network after removing the node i.

After determining the Laplacian centrality of each node using the weighted matrix, each node's label influence will be equal to the central value of that node, which is defined in Eq. (7) as follows:

$$\mathrm{LI}(i, l) = [\mathrm{LAPC}]_i \tag{7}$$

LI (i, l) represents the influence of the label l on the node i, in which $[\mathrm{LAPC}]_i$ is a laplacian centrality of a node(i).

Definition 5 (*Label Acceptance (LA)*). In the original LPA, all neighbors have the same probability of propagating a label. In the SAS-LP method, label influence (LI) of neighbors is used to select the best label for propagation. Label Acceptance is computed by Eq. (8) as follows:

$$\mathrm{LA(i)} = \left[argmax LI_{v\in\Gamma(u)}(v, l) \right] \tag{8}$$

$\Gamma(u)$shows the first-order neighborhood of node $u \in V$. In the stage of updating the labels, every single node will get the node label with considerable influence among its first-degree adjacent nodes.

3.2. Proposed SAS-LP algorithm

By assuming an attributed graph $G=(V, E, A)$ and the number of clusters K, the propose of clustering problem studied in the present investigation is to partition the node-set V of G into K disjoint subsets $V_1, V_2, \ldots, V_n$, where $V = \bigcup_{i=1}^{n} v_i$ and $V_i \cap V_j = \varnothing$ for any $i \neq j$, so that: (1) the nodes within clusters are densely connected with regard to structure, while the nodes in different clusters are sparsely connected; and (2) the nodes within clusters have low diversity in their attribute values with regard to attribute, while the nodes in different clusters may have diverse attribute values. The LPA algorithm, which uses only structural information of graph for community detections, initially assigns a unique label for every node and subsequently selects the node with the highest frequency in several updated stages. If the algorithm reaches an iteration that the label of each node is equal to the maximum number of adjacent tags in the node and no longer changes occur, all nodes under the dense subgraph that have reached the same label are detected as community graphs. However, this algorithm faces instability and low performance due to the development of monster communities resulting from the equal importance of nodes and random behavior in the updating phase and tie-break mode. In each cluster, there are more important vertices that make a significant contribution (the center of cluster), so that one gets closer to the center of the community from its boundary, more important vertices (with more significant influence) emerge; therefore, the higher the influence of a vertex in a graph compared to its adjacent graph can indicate the more significant role of that vertex. The nodes in the community have different influence rankings. The nodes with greater influence play the role of the dominator, and the nodes with lower influence play the role of subordinator. One node can affect other nodes (dominator) or be affected by other nodes (subordinator), and the importance of all nodes in the cluster is not similar.

The objective of the present study is to detect these nodes by influencing the structure and attributed dimensions to identify clusters in the attributed graph. For this purpose, two new concepts are introduced and used by the authors in high-precision algorithm LPA to solve the problem of attribute clustering. In the first concept, a new similarity will be defined for both nodes according to the combination of structure and attribute. The weighted graph is obtained using this similarity, and the influence of each node based on the Laplacian centrality in a weighted graph is calculated by the authors through employing the second concept. In the following, the graph $G = (V, E, A)$ will be converted to the graph $G = (V, E, W)$ with structure-attributes fusion using the Eq. (3), which is the weight of the edge between two nodes. Afterward, using Laplacian centrality in the weighted graph, the centrality of each node will be calculated, which will be equal to the influence of that node; the influence of nodes on their adjacent will not be only in respect of structure but also in terms of attributes. The nodes with higher Laplacian centrality in the weighted matrix presented in Eq. (3) will be permeable on their adjacent in terms of structure and attributes. The nodes that possess a central position in the clusters and have a considerable number of connections with other members of the group make a significant contribution in controlling, guiding, and establishing strength and stability in the community, while the nodes that are on the boundary of communities may lead and guide an intermediary role between communities. The attributed graph consists of the adjacency matrix and the attributes matrix. Based on these matrices, the authors will describe the structural similarity matrix and the attribute similarity matrix based on some formulas. Also, the authors will use some

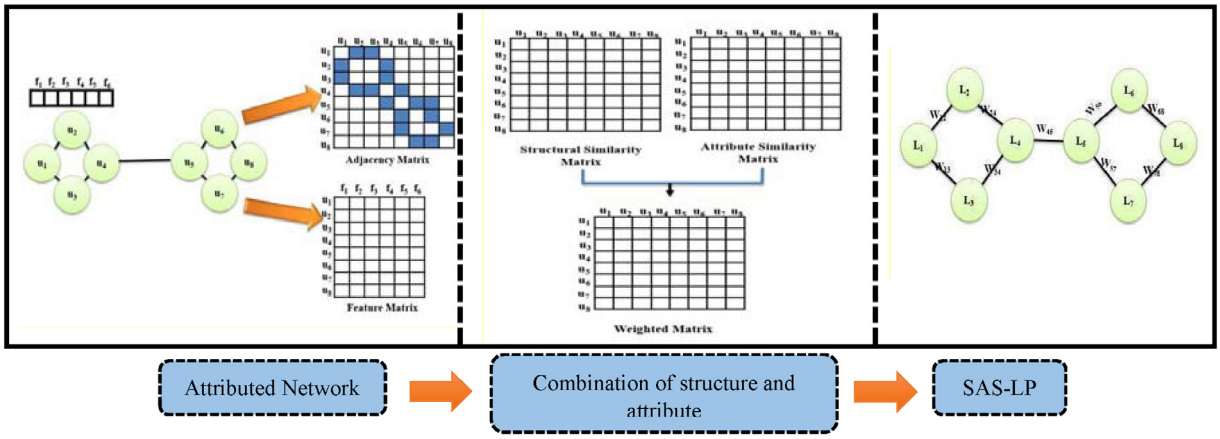

Fig. 2. The steps of SAS-LP algorithm.

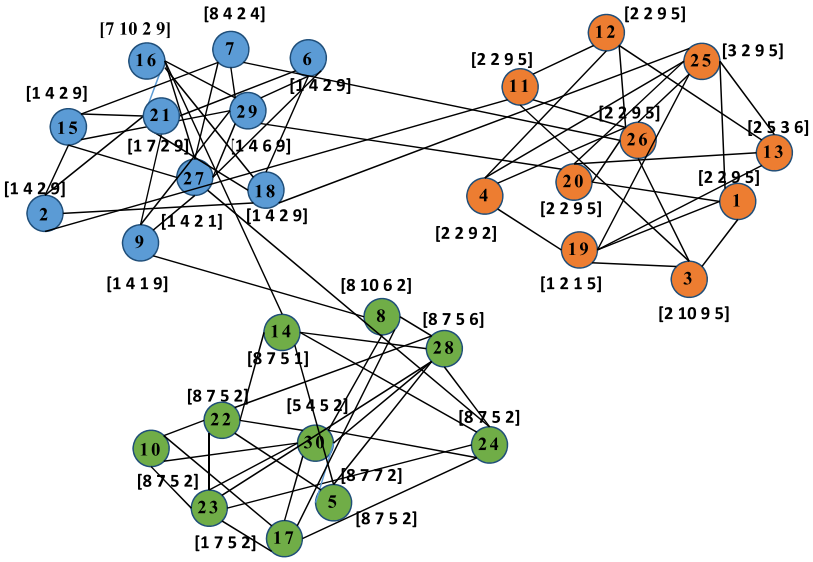

Fig. 3. Graph structure and ground truth for the LFR-EA dataset with mixing parameter (μ) = 0.1 and attribute noise (v) = 0.1: the 30 nodes are partitioned into 3 distinct communities.

formulas to produce the weighted matrix. The produced weighted matrix will be applied to calculate the influence of the nodes using Laplacian centrality. Moreover, the selection and updating of the nodes will be performed based on the algorithm and some formulas. The authors will sum the node labels and the influence of nodes and select the label with greater influence. The tie-break mode occurs very infrequently in the algorithm proposed in the present paper; hence, the link weights will be considered, and the authors will sum the weight links of similar labels and select the largest of them. The steps of the algorithm are presented in Fig. 2.

Example 1. Consider the network LFR-EA with mixing parameter = 0.1 and attribute noise = 0.1 shown in Fig. 3, which indicates the update process of the SAS-LP method (Algorithm 1). All the steps of this procedure are explained in Table 1, which the authors have considered in more detail following. In the beginning, all nodes are uniquely labeled similar to LPA. Then, the weighted similarity and label influence of nodes (Laplacian centrality) are computed. The most significant label influence is considered as the first node for updating its label. This node updates its label with the selected label from its neighbors with the highest label influence. For instance, by employing SA-LPA, the updating order is 30 → 26 → 23 → 24 → 17 → 21 → 15 → 6 → 18 → 20→ . . . → 19 → 17 → 13 sequences. Node 30 is firstly updated with the label of node 23 since it has a more considerable label influence for node 30 among its neighbors. After node 30, node 26 is updated. It updates its label with the label of node 20 because of the more significant label influence of node among its neighbors. Then, node 23 is updated with label 23. All other nodes similarly update their label according to their label influence value. At the end of iteration 1, nodes blue, green, and are gathered in the same community because they received the same label 23, 20, 15 nodes 7. Also, nodes 9 to 18 are collected in another community because their label is updated with the label of node 16.

Table 1
Iterations of the algorithm SAS-LP based on Fig. 3.

		Iteration 1			Iteration 2			Iteration 3		
Node	Label Influence	Order updating	Current label	New label	Order updating	Current label	New label	Order updating	Current label	New label
1	28	30	30	23	30	23	23	30	23	23
2	30	26	26	20	26	20	20	26	20	20
3	24	23	23	23	23	23	23	23	23	23
4	23	24	24	23	24	23	23	24	23	23
5	18	17	17	23	17	23	23	17	23	23
6	35	21	21	15	21	15	15	21	15	15
7	11	15	15	15	15	15	15	15	15	15
8	27	6	6	15	6	15	15	6	15	15
9	16	18	18	15	18	15	15	18	15	15
10	10	20	20	20	20	20	20	20	20	20
11	30	25	25	20	25	20	20	25	20	20
12	31	12	12	20	12	20	20	12	20	20
13	3	2	2	15	2	15	15	2	15	15
14	16	29	29	15	29	15	15	29	15	15
15	39	11	11	20	11	20	20	11	20	20
16	14	10	10	23	10	23	23	10	23	23
17	43	1	1	20	1	20	20	1	20	20
18	33	27	27	15	27	15	15	27	15	15
19	12	28	28	23	28	23	23	28	23	23
20	33	8	8	23	8	23	23	8	23	23
21	42	3	3	20	3	20	20	3	20	20
22	23	4	4	20	4	20	20	4	20	20
23	58	22	22	23	22	23	23	22	23	23
24	51	5	5	23	5	23	23	5	23	23
25	32	9	9	15	9	15	15	9	15	15
26	60	14	14	23	14	23	23	14	23	23
27	28	16	16	15	16	15	15	16	15	15
28	28	19	19	20	19	20	20	19	20	20
29	30	7	7	15	7	15	15	7	15	15
30	82	13	13	20	13	20	20	13	20	20

3.3. Pseudocode

Algorithm 1: The proposed SAS-LP community detection algorithm

Input: network $G=(V, E)$
Output: Community structures $C = \{C_1, \ldots .. C_k\}$
1. converting the attribute graph into a weighted graph
2. Assign a unique label to each node in the network
3. Calculate the label influence by Eq. (7)
4. **While** the label of nodes change or $t <$ Max iteration **do**
5. Arrange node in ascending order of node strength and put the results on the vectorχ.
6. Set $t = 1$.
7. For each node $v_i \in X$ vector, update its labels according to the acceptance label in Eq. (8)
8. If a tiebreak state is happening, calculating the sun label of a neighbor node and select the label has a higher node's strength.
9. $t = t + 1$.
10. **End while**
11. Construct communities based on a similar label.
12. **Return** community structures.

3.4. Computational complexity

The time complexity of the proposed SAS-LP algorithm is discussed in this section. Assume that a network with $n=|V|$ nodes and $m=|E|$ links $G=(V, E)$, suppose k to be the average node degree. This algorithm consists of several isolated steps. Each step individually runs on different time complexity. In the first stage, the time complexity for initializing all nodes with unique labels is denoted by $O(n)$. The second step, the calculation of label influence. For calculating label influence, the attribute similarity and structural similarity should be calculated, the time complexity of each of them is $O(nk)$, that is equal to $O(m)$ because complex networks usually possess a massive graph with a small number of neighbors, and Laplacian centrality is computed as $O(m)$. The third step is, ranking the nodes based on label influence that possesses the time complexity of $O(n)$ (due to the possibility of using radix and bucket sorting algorithm in a liner time). The fourth step is the label propagation process. The time complexity of label update is also computed in $O(nk)$ according to the acceptance label due to considering only neighbors that are equal to $O(m)$. Finally, $O(n)$ is the time complexity of assigning the nodes with the same label to its community. In general, $O(3nk + m + 2n) \approx O(m)$ the time complexity of the proposed algorithm is. Since scale-free networks have sparsity property, the number of edges is approximately equal to the number of nodes, hence, $O(m) \approx O(n)$.

4. Experimental evaluation

In this section, the experiment results for SAS-LP are presented. A series of experiments are conducted to evaluate the proposed method of performance comprehensively. The organization is as follows. Section 4.1, summarizes all the datasets are used in the following experiments. The effectiveness of the method is evaluated on two groups of datasets: 1) synthetic and 2) real-world datasets. Section 4.2 reviews evaluation metrics. Section 4.3 discusses the comprehensive results of the synthetic datasets, including a parameter analysis experiment and the general evaluation for

the capacity of the proposed method. Section 4.4 presents the results of the performance evaluation on real-world datasets compared with several state-of-the-art methods, and Section 4.5 shows the relationship between community centers and Laplacian centrality. All the experiments were carried out in a desktop pc equipped with a quad-core Intel i7 2.20 GHz processor and 16 GB RAM.

4.1. Datasets

To validate and assess the performance of the SAS-LP, two class of datasets are used. The synthetic dataset is computer-generated networks allowing the creation of the ground truth useful to evaluate the similarity between the synthetically generated and the detected communities. The real-world datasets, extracted from real environments, better represent the actual network behavior. The description of these networks is as follows.

4.1.1. Synthetic dataset

LFR-EA is the synthetic network which is generated using the benchmark proposed by Elhadi and Agam (Liu and Lü, 2010). It is an extension of the LFR benchmark of Lancichinetti et al (Lancichinetti et al., 2008). The network generator uses two parameters μ and ν, both ranging in the interval [0.1, 0.9], to control the structure and attribute values, respectively. The mixing parameter μ determines the rate of intra and inter-community connections. Low amounts of μ give a clear community structure where the intracluster link is much more than inter-cluster links. Analogously ν is the noise attribute parameter in which low values generate similar features of nodes belonging to the same community. The combination of μ and v values produces graphs with a clear to ambiguous structure and/or attributes.

4.1.2. Real-World datasets

Cora (Sen et al., 2008) contains a set of nodes representing scientific publications, where an edge between two nodes is a citation from a publication to another. The attributes' domain of this network is represented by a set of unique words. If a word is present in this paper, the attribute for that word is set to 1, 0 otherwise. Each node has been classified into seven classes: 1) case-based reasoning; 2) genetic algorithms; 3) neural networks; 4) probabilistic methods; 5) reinforcement learning; 6) rule learning; and 5) theory.

Citeseer (Sen et al., 2008) is another citations network where each node belongs to one of the following six categories: 1) agents; 2) artificial intelligence; 3) databases; 4) human–computer interaction; 5) information retrieval; and 6) machine learning.

WebKB Dataset (Sen et al., 2008) consists of scientific publications, which include Web page networks of four universities: 1) Cornell; 2) Texas; 3) Washington, and 4) Wisconsin. Each page network can be classified into five classes: 1) course; 2) faculty; 3) student; 4) project; and 5) staff.

Political Blogs Dataset is a network of Weblogs on U.S. politics with hyperlinks between these Weblogs. Each Weblog is associated with an attribute describing the political leaning of the Weblog, labeled as either liberal or conservative. The statistical features of these test networks are summarized in Table 2.

4.2. Evaluation protocol

Two main groups of metrics are used to evaluate the performance of community detection methods, called external and internal measures. While the latter is often used when the true labels are not accessible. In the following, these indices will be described.

4.2.1. NMI

The normalized mutual information NMI (*A, B*) of two divisions *A* and *B* of a network is defined as follows. Let *C* be the confusion matrix whose element C_{ij} is the number of nodes of community *i* of the partition *A* that are also in the community *j* of the partition *B*.

$$NMI(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log(nC_{ij}/C_iC_j)}{\sum_{i=1}^{C_A}C_i.log(C_i/n)+\sum_{j=1}^{C_B}C_j.log(C_j/n)} \tag{9}$$

where $C_A(C_B)$ is the number of groups in the partition *A(B)*, $C_i(C_j)$ is the sum of the elements of *C* in row *i* (column j), and n is the number of nodes. If *A* = *B*, NMI (*A*, *B*) = 1. If *A* and *B* are completely different, NMI *(A, B) = 0*.

Accuracy is a common statistical measure that refers to the closeness of the measurements to a specific value. In the community detection, Accuracy means better correspondence between the communities extracted and the groups in the real network. It represents the ratio of the number of correct clustering nodes to all nodes.

4.2.2. F1-score

F1-score captures the level of approximation reached by network partitions obtained through community discovery algorithms w.r.t. Ground-truth ones. Moreover, it allows for a visual inspection of the partition quality exploiting density scatter plots.

4.2.3. Density

Strong connection among vertices is analyzed by using the density function, which represents the ratio between the number of edges presented in the clusters and the total number of edges in the whole graph. The ratios get accumulated for all clusters to evaluate the overall impact. Density values lie in the interval of [0, 1].

$$\delta\left(\{C\}_{i=1}^{k}\right)=\frac{1}{\|E\|}\sum_{i=1}^{k}\|E(C_i)\| \tag{10}$$

where $\|E\|$and, $\|E(C_i)\|$ are denoted as the number of edges in the graph and each cluster, respectively.

4.2.4. Entropy

One of the key aspects to measure the quality of clustering results is to determine the relevancy among vertices based upon their attributed nature. For each attribute, the entropy is calculated against each cluster with associated attributes. When all the vertices inside the same cluster are having similar attributes or contexts associated with them, then overall entropy acquires minimum value.

$$Entropy(a_t)=\sum_{i=1}^{k}\frac{\|C_i\|}{\|V\|}entropy(a_t,C_k) \tag{11}$$

$$entropy(a_t,c_k)=-\sum_{s=1}^{\|dom(a_t)\|}p_{ks}^{t}logp_{ks}^{t}$$

Table 2
Features of the real-world datasets.

Dataset	Nodes	Edges	Attributes	Community
Citeseer	1787	3285	3703	6
Cora	2708	5429	1433	5
Cornell	195	304	1703	5
Texas	187	328	1703	5
Washington	230	446	1703	5
Wisconsin	265	530	1703	5
Political Blogs	1490	19,090	7	2

Table 3
LFR-EA-1000 parameters setting.

Parameter	Value
Number of nodes (N)	1000
Average Degree(k)	25
Maximum degree (maxk)	40
Mixing parameter (μ)	[0.1;0.9]
Exponent for the community size distribution (t1)	1
Minimum for the community size (minc)	60
Maximum for the community size (maxc)	100
Number of overlapping nodes (om)	0
Number of attributes	4
Attribute's domain cluster assignment (ainf)	1
Attribute range (R)	10
Attribute noise	[0.1;0.9]

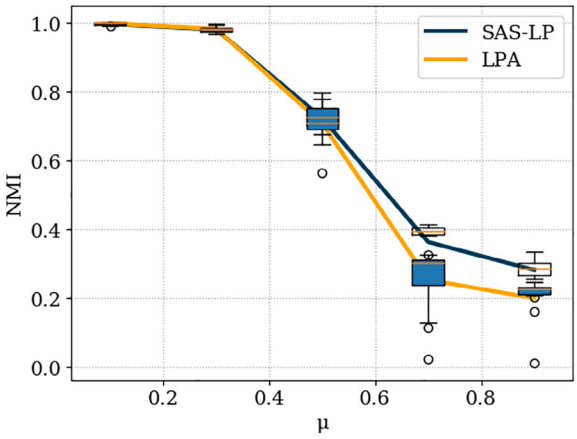

Fig. 4. NMI Values for LPA and SAS-LP with different mixing parameter by setting ν = 0.1.

where p^{t}_{ks} is the fraction of vertices in cluster C_k that take the value s where $s \in \text{dom}\ (a_t)$.

4.3. Evaluation on synthetic datasets

The authors generated a benchmark of networks consisting of 1000 nodes, named LFREA-1000, to evaluate all aspects of SAS-LP. Different instances of the combination of parameters reported in Table 3 are generated. Since generating networks are the stochastic procedure and different runs may lead to different resulting partitions, so we average the results over ten runs.

1. **Mixing parameter** μ: For the case of varying μ, the attribute noise is fixed v = 0.1. Fig. 4 shows the performance of SAS-LP on generated attributed graphs. In this examination, the proposed method is compared with the original LPA to demonstrate the effectiveness and stability of it. Compared with LPA, both SAS-LP and LPA are perform great in well-structured communities. As the mixing parameter goes up, the stability and performance of the LPA decrease, while SAS-LP keeps stability in every step. Furthermore, the proposed method has a comparable performance when the graphs have an ambiguous community structure. As the graph structure becomes less clear, employing attributed plays a vital role. Thus, the capability of the SAS-LP is due to exploiting both graph structure and attributes linearly.
2. **Noise parameter v**: Fix the μ parameter, SAS-LP is compared with different noise setting. To this end, the authors fixed the mixing parameter to μ=0.5 in order to have an attributed graph with the structure that is sufficiently ambiguous. The performance comparison of the proposed method is summarized in Table 4. These results demonstrate the robustness of the algorithm to the noise. As can be seen from Table 4, all results except entropy remain stable. As already outlined, low entropy values indicate homogeneous communities from the attributes perspective, and the results due to the noise confirm this evidence. Density measures the connectivity around vertices. From the data in Table 4, the authors have concluded that adding vertices attribute promotes the performance of community detection in most cases.
3. **nattr range parameter**: To assess the quality of the results of the attribute range obtained by SAS-LP, the authors report the results in different noise span (Fig. 5). In particular, with the increasing range of attribute values, entropy values go up by increasing the noise parameter. It is worth pointing out that entropy not only sensitive to the noise parameter but also sensitive to the range of attribute values. The same conclusion also holds for the NMI, and this is due to the fact that a broad range of attributed leads to less similar nodes. Intuitively, when employing all types of information and match them significantly, it will let have a stable range of accuracy while existing some noise in the data.
4. **nattr domain size**: In this experiment, the authors attempt to analyze the impact of domain size. As can be seen from Fig. 6, the SAS-LP can handle all tested domain size, and the performance of the algorithm remains stable. The reason for this is that (1) the simple match coefficient index does not sensitive to the number of attributes; (2), using the composition of attribute similarity and structure similarity leads to more accurate and stable results.
5. **α parameter**: For Fair comparison in this experiment, the results are reported in two settings of node attribute nose v = 0.1, v = 0.3 (Fig. 7). It is worth pointing out that the choice of using topological information alone is not sufficient to find a good clustering. These results confirm the importance of considering both structural and attribute components to obtain high-quality partitions. In conclusion, by adding the adaptive parameter alpha to control the trade-off between structure and attribute, SAS-LP robustly combines such two sources of information and maintain high-level NMI, even there is an ambiguous structure exists.
6. **The number of nodes (N)**: To evaluate the effect of scalability, the authors compare the proposed method with different network sizes on both external and internal measures. From

Table 4
Comparison Results on Different noise Parameter by setting *mu* = 0.5.

Noise	NMI	ACC	F1	Entropy	Dens
0.1	0.7241	0.5649	0.5734	0.3676	0.3991
0.3	0.7304	0.5853	0.5978	0.5241	0.3989
0.5	0.7122	0.5614	0.5726	0.6628	0.4000
0.7	0.7165	0.5398	0.5594	0.7396	0.6530
0.9	0.7175	0.5600	0.5761	0.7820	0.3962

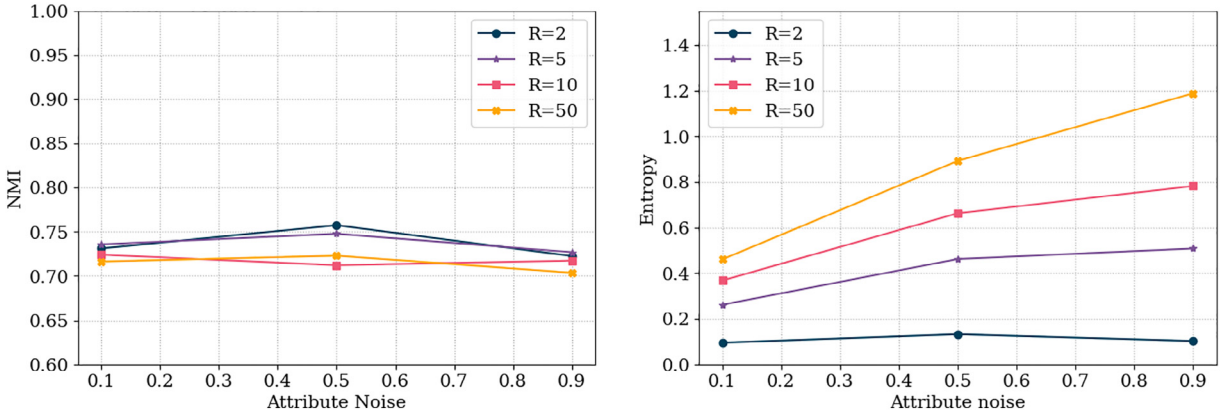

Fig. 5. NMI and Entropy Values for Attribute range by setting μ = 0.5.

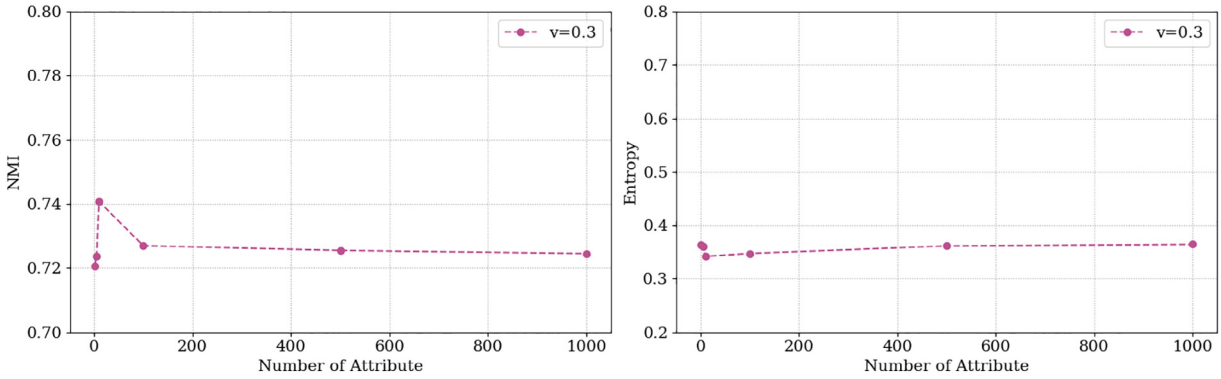

Fig. 6. NMI and Entropy Values for Attribute domain by setting μ = 0.5.

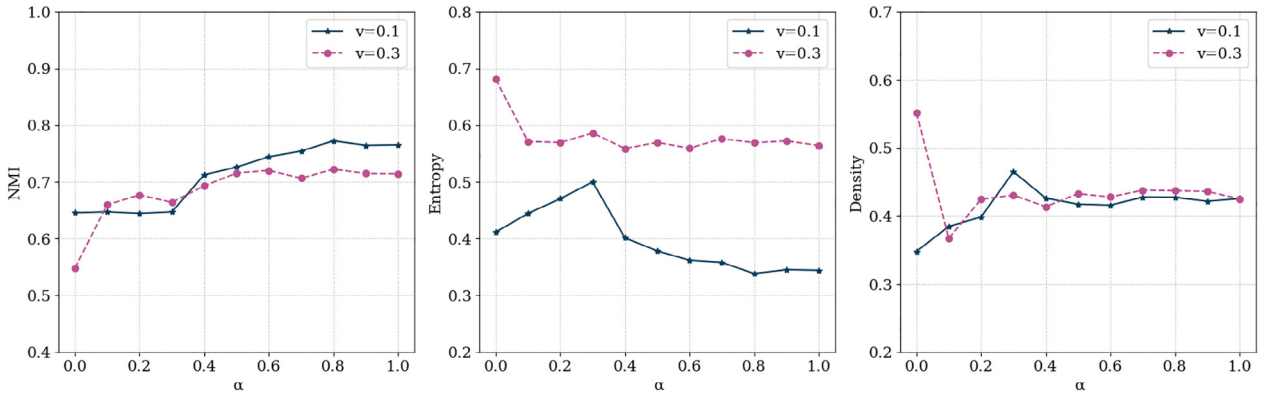

Fig. 7. NMI, Entropy and Density Values for α parameter by setting μ = 0.5.

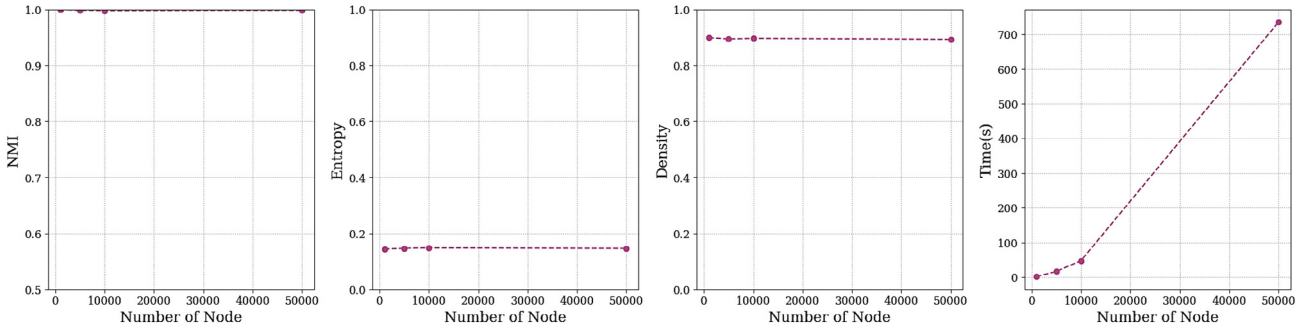

Fig. 8. NMI, Entropy, Density and Time Values for Number of Nodes by setting μ = 0.1.

Fig. 8, it is observed that the NMI of experimental results on various networks size is optimal in all cases. Besides, the authors can clearly see that the connectivity between vertices in a cluster remains stable in different networks size. Additionally, from Fig. 8, it is easy to notice that largely ascribes to its linear rate of convergence. The results indicate that SPA-LP's performance remains high in the large-scale complex network. Also, entropy had not sensible change and indicate the robustness of the proposed algorithm in high dimensions.

4.4. Evaluation of real-world datasets

4.4.1. Baseline methods

The authors compared a total of nine algorithms with the method in the experiments. The graph clustering algorithms include approaches that use both node attributes and network structure information.

Adapt-SA (Li et al., 2017) is a weighted K-means algorithm with local learning for attributed graph clustering.
Adapt-SA-soft (Li et al., 2017) is an extended Adapt-SA which has an additional step to the fuzzy K-means.
Adapt-SA(PCA) (Li et al., 2017) using PCA for dimensionality reduction.
SA-Cluster (Zhou et al., 2009) performs matrix multiplication to calculate the random walk distances between graph vertices, and edges weights are iteratively adjusted to balance the importance between structural and attribute similarities.
Inc-cluster (Zhou et al., 2010) is extended of SA-Cluster, which incrementally updates the random walk distances given the edge weight increments.
PCL-DC (Yang et al., 2009) is a discriminative approach for modeling the contents of nodes via a probabilistic framework through the shared variables of community member-ships with both link and content information.
BAGC (Xu et al., 2012) is a Bayesian probabilistic model, which provides a principled and natural framework for capturing both structural and attribute aspects of a graph while avoiding the artificial design of a distance measure.
PPSB-DC (Chai et al., 2013) is a popularity-productivity stochastic block, which explicitly exploits the popularity and productivity of nodes to model the differences of nodes in receiving links and in producing links.

4.4.2. Experimental analysis of edges weights

In this section, to explicitly clarify the effectiveness and reasonability of edges and node weights. SAS-LP is assessed with different evaluation criteria on the Citeseer, Cora, and WebKB data sets shown in Fig. 9. The authors examine each dataset by assigning different alpha, ranging from 0 to 1 with the step length 0.1, where 1 indicated that only the node attribute information of a network was utilized. With the decrease of alpha, more structure information was taken into account. The horizontal axis is the alpha range.

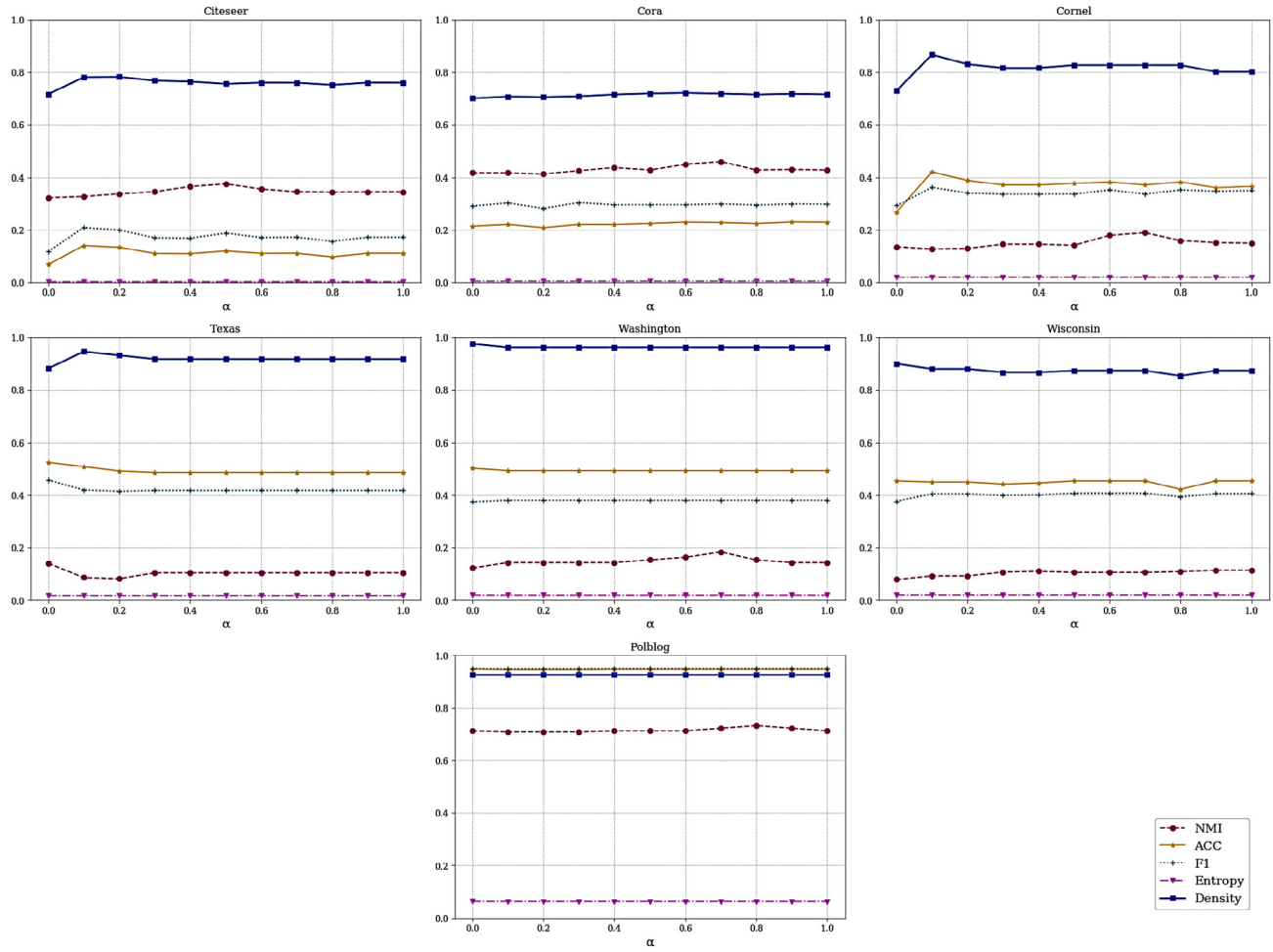

Fig. 9. Performance Values for α parameter. In data-set of Citeseer, Cora, Cornel, Texas, Washington, Wisconsin, and Polblog.

As nodes are sorted by their assigned weights, it does not need to run multiple times. As shown in Fig. 9, the authors observe that results on using both the structure and content information of the graph generally perform better than those using only one side of information. In all datasets except Texas, the results of the usage of both sides of information are quite tangible. In these networks, there is a pic in the middle of the plot which admits using both sides of information increase the NMI values. Texas network, due to the asteroid shape, tends to capture more structure information rather than node attributes. Additionally, it is easy to notice that the proposed method can homogeneously cluster nodes. The entropy value almost near to zero in all of the datasets. As outlined, the density and entropy values, the former considering the internal density of the partitions and the latter the attribute homogeneity and low entropy means groups with similar objects. For Cora and Citeseer networks, the model gets better results than the other

Table 5
Experimental results on real datasets.

		NMI (↑)	ACC (↑)	F1(↑)	Time (s ↓)
Cora	Adapt-SA	0.454 ± 0.010	0.615 ± 0.006	0.485 ± 0.002	33.68
	Adapt-SA-soft	0.441 ± 0.016	0.582 ± 0.009	0.440 ± 0.008	34.45
	Adapt-SA(PCA)	0.203 ± 0.001	0.384 ± 0.006	0.277 ± 0.000	13.06
	SA-Cluster	0.117 ± 0.000	0.264 ± 0.000	0.282 ± 0.000	5.16
	Inc-cluster	0.112 ± 0.000	0.267 ± 0.000	0.284 ± 0.000	4.90
	PCL-DC	0.416 ± 0.003	0.564 ± 0.009	0.441 ± 0.002	5808.69
	BAGC	0.008 ± 0.005	0.301 ± 0.007	0.299 ± 0.006	4.01
	PPSB	0.068 ± 0.035	0.263 ± 0.006	0.190 ± 0.010	20116.21
	PPSB-DC	0.466 ± 0.028	0.620 ± 0.032	0.512 ± 0.021	23516.71
	SAS-LP	0.471 ± 0.000	0.631 ± 0.000	0.497 ± 0.000	3.38
Citeseer	Adapt-SA	0.388 ± 0.017	0.621 ± 0.010	0.488 ± 0.009	52.95
	Adapt-SA-soft	0.293 ± 0.002	0.541 ± 0.004	0.396 ± 0.007	80.29
	Adapt-SA(PCA)	0.231 ± 0.012	0.476 ± 0.012	0.348 ± 0.010	9.09
	SA-Cluster	0.047 ± 0.000	0.233 ± 0.000	0.298 ± 0.000	10.47
	Inc-cluster	0.043 ± 0.000	0.230 ± 0.000	0.299 ± 0.000	12.75
	PCL-DC	0.170 ± 0.003	0.412 ± 0.016	0.299 ± 0.002	1831.04
	BAGC	0.017 ± 0.000	0.222 ± 0.000	0.298 ± 0.000	10.94
	PPSB	0.033 ± 0.001	0.263 ± 0.006	0.190 ± 0.005	22798.67
	PPSB-DC	0.387 ± 0.015	0.619 ± 0.030	0.512 ± 0.005	35778.21
	SAS-LP	0.395 ± 0.000	0.614 ± 0.000	0.491 ± 0.000	2.27
Cornell	Adapt-SA	0.168 ± 0.022	0.437 ± 0.008	0.389 ± 0.005	0.21
	Adapt-SA-soft	0.134 ± 0.030	0.452 ± 0.016	0.378 ± 0.023	0.27
	Adapt-SA(PCA)	0.122 ± 0.011	0.444 ± 0.003	0.362 ± 0.002	0.02
	SA-Cluster	0.064 ± 0.000	0.415 ± 0.000	0.386 ± 0.000	1.33
	Inc-cluster	0.038 ± 0.000	0.405 ± 0.000	0.401 ± 0.000	1.46
	PCL-DC	0.073 ± 0.010	0.329 ± 0.014	0.281 ± 0.000	289.91
	BAGC	0.040 ± 0.006	0.439 ± 0.003	0.342 ± 0.034	1.03
	PPSB	0.068 ± 0.001	0.362 ± 0.026	0.308 ± 0.007	1145.66
	PPSB-DC	0.121 ± 0.001	0.536 ± 0.010	0.477 ± 0.015	2355.98
	SAS-LP	0.241 ± 0.000	0.550 ± 0.000	0.440 ± 0.000	0.02
Texas	Adapt-SA	0.294 ± 0.060	0.619 ± 0.023	0.561 ± 0.010	0.32
	Adapt-SA-soft	0.221 ± 0.057	0.559 ± 0.039	0.493 ± 0.027	0.38
	Adapt-SA(PCA)	0.135 ± 0.056	0.550 ± 0.011	0.538 ± 0.020	0.02
	SA-Cluster	0.082 ± 0.000	0.401 ± 0.000	0.383 ± 0.000	1.22
	Inc-cluster	0.106 ± 0.000	0.423 ± 0.000	0.399 ± 0.000	1.37
	PCL-DC	0.061 ± 0.011	0.348 ± 0.015	0.316 ± 0.018	134.26
	BAGC	0.052 ± 0.007	0.563 ± 0.003	0.546 ± 0.003	0.97
	PPSB	0.111 ± 0.015	0.506 ± 0.012	0.467 ± 0.005	1543.36
	PPSB-DC	0.305 ± 0.002	0.629 ± 0.015	0.605 ± 0.015	2323.71
	SAS-LP	0.343 ± 0.000	0.597 ± 0.000	0.621 ± 0.000	0.02
Washington	Adapt-SA	0.342 ± 0.032	0.628 ± 0.032	0.582 ± 0.0230	0.34
	Adapt-SA-soft	0.261 ± 0.045	0.572 ± 0.044	0.518 ± 0.046	0.41
	Adapt-SA(PCA)	0.215 ± 0.012	0.567 ± 0.018	0.532 ± 0.023	0.02
	SA-Cluster	0.077 ± 0.000	0.491 ± 0.000	0.474 ± 0.000	1.86
	Inc-cluster	0.063 ± 0.000	0.465 ± 0.000	0.472 ± 0.000	1.95
	PCL-DC	0.092 ± 0.015	0.380 ± 0.039	0.326 ± 0.034	136.04
	BAGC	0.053 ± 0.006	0.464 ± 0.003	0.480 ± 0.002	1.23
	PPSB	0.112 ± 0.006	0.402 ± 0.021	0.358 ± 0.009	1736.32
	PPSB-DC	0.239 ± 0.021	0.571 ± 0.012	0.498 ± 0.020	2531.55
	SAS-LP	0375 ± 0.000	0.656 ± 0.000	0.550 ± 0.000	0.02
Wisconsin	Adapt-SA	0.330 ± 0.046	0.560 ± 0.017	0.498 ± 0.029	0.11
	Adapt-SA-soft	0.323 ± 0.025	0.551 ± 0.009	0.485 ± 0.024	0.53
	Adapt-SA(PCA)	0.108 ± 0.031	0.509 ± 0.029	0.456 ± 0.028	0.05
	SA-Cluster	0.101 ± 0.000	0.404 ± 0.000	0.398 ± 0.000	1.54
	Inc-cluster	0.089 ± 0.000	0.464 ± 0.000	0.426 ± 0.000	1.74
	PCL-DC	0.060 ± 0.000	0.336 ± 0.000	0.274 ± 0.002	103.56
	BAGC	0.034 ± 0.015	0.474 ± 0.011	0.479 ± 0.005	1.37
	PPSB	0.078 ± 0.013	0.385 ± 0.032	0.328 ± 0.006	1403.33
	PPSB-DC	0.232 ± 0.031	0.493 ± 0.016	0.421 ± 0.022	2908.72
	SAS-LP	0.358 ± 0.000	0.572 ± 0.000	0.501 ± 0.000	0.05

dataset on NMI. The reason is that the structures in the two data sets are assortative. As mentioned, before the NMI describes the quality of similarity between partitions. However, the other datasets get better results on Accuracy and f-score. In those datasets, SAS-LP can assign more correct labels to the nodes. The above examines the real data sets strongly convince that applying both the graph structure and node attribute contains useful information for community detection and illustrates the significance of capturing the interplay between two-sides' information. It is worth mentioning that one measure is not able to discriminate against the effectiveness of the model.

4.4.3. Comparative result

In this section, to better prove the efficiency, the results of the approach are compared to some recent methods which use both structure information and node attributes. The baseline methods are described in Section 4.4.1. The clustering results are shown in Table 5. To measure the clustering result, the authors employ NMI, Accuracy, and f1-score metrics and present values in the tables. The authors did not report the entropy value due to the entropy vanishing evidence. It almost vanishes and does not change its values significantly for different fusion coefficients. This is probably because of the sparsity of the attributes under consideration so that there is a big part of nodes with similar zero attributes in each cluster, making entropy vanishing (Chunaev et al., 2019). From these six tables, the authors can find that the SAS-LP achieves significant improvement compared with plain network clusterisation approaches and beats other attributed network clusterisation approaches in most situations. From the results the Table 5, the authors can find out the proposed SAS-LP verifies the effectiveness of community detection, handling different network structure complexity, which highlights its strengths. In an associative network like Cora and Citeseer, the quality of clustering is higher than the other state of the art methods. PPSB-DC and Adapt-SA in some of the datasets like Texas have higher accuracy while their NMI remains low. This evidence happened in unbalanced datasets where each partition has a various number of nodes. In this situation, if assign all nodes to the label of maximum length partition, higher, it achieves high accuracy while it cannot be clustered effectively. Besides, In Timing, SAS-LP outperforms other methods on all datasets. This demonstrates that the proposed method is linearly exploring through the graph and label each node. Summarizing, real-world datasets quite admit the results on synthetic datasets in Section 4.3.

4.5. Correlation between community cores and Laplacian centrality

In this section, the authors intend to show that the centrality of the community core in the attributed graph, with higher propagation influence power, possesses a higher Laplacian centrality compared to other nodes. First, the dataset will be generated with parameters with mixing parameter = 0.1, Number of nodes = 1000, Average Degree = 5, Maximum degree = 25, Exponent for the node degree = 2, Exponent for the community size = 1, the minimum size of community = 30, the maximum size of community = 50, Number of attributes = 6, and Attribute noise = 0.1. The ground-truth form of this data set possesses 25 communities, as indicated in Fig. 10. Figs. 11–13 show the scattering of nodes on the graph presented in Fig. 10, based on the three Laplacian, PageRank, and Degree centralities. The proposed algorithm in the present paper correctly identifies 25 communities. At the center of these 25 communities, there are some nodes the labels of which have remained relatively constant during the algorithm's implementation steps (except the first step, the label will change, and in the next steps, the label will remain unchanged, and the other node labels will be changed to their node label). In the last stage of the proposed algorithm, there are 25 labels left, which are related to the 25 labels of the core of

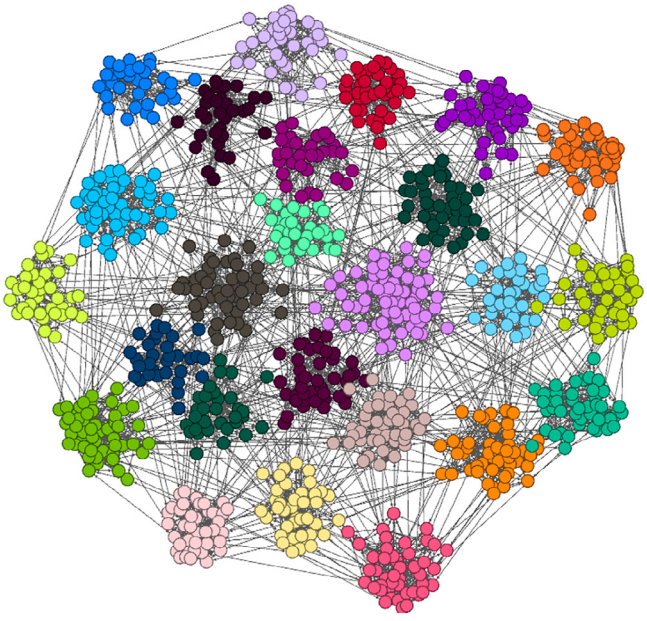

Fig. 10. LFR-EA dataset N = 1000, μ = 0.1, and v = 0.1.

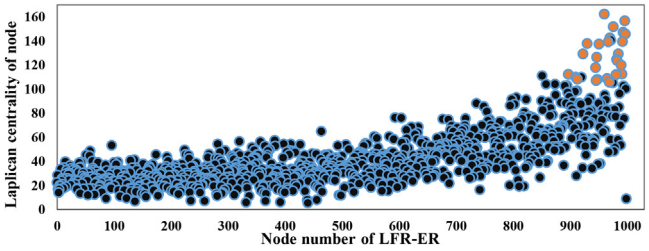

Fig. 11. Distribution of nodes Laplacian centrality in LFR-ER (1000).

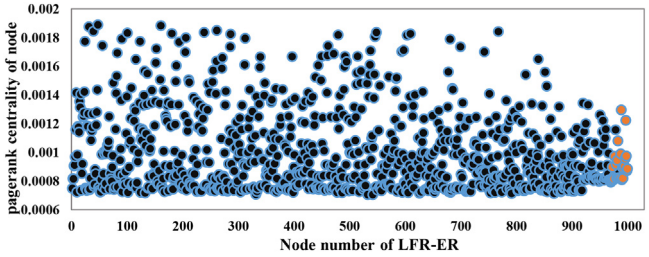

Fig. 12. Distribution of nodes PageRank centrality in LFR-ER (1000).

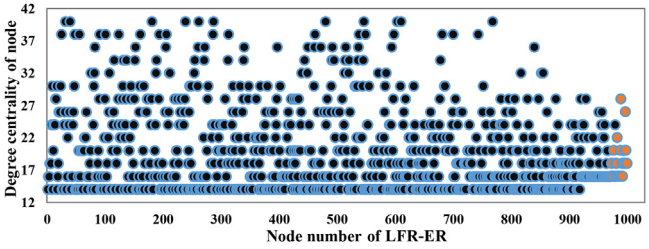

Fig. 13. Distribution of nodes degree centrality in LFR-ER (1000).

these communities. These core nodes, with high propagation influence and the ability to change the labels of the rest of the nodes to their labels, are nodes that have a high laplacian center. All the orange nodes in Fig. 11 have a high numerical value of laplacian. Although these nodes have the center of the communities, which is shown in Figs. 12 and 13, with low page rank and degree centrality. Therefore, it can be concluded that in the attributed graph, the nodes of the central core, with higher emission powers, have higher laplacian centralities. Therefore, it can be concluded that the nodes with high centrality of laplacian were the nodes of the core of the communities with high penetrations.

5. Conclusion

One of the emergent properties that occur in the middle layer of complex networks is the presence of clusters in which the dense of edge with each other is more than in other areas. The detection of this dense subgraph in attributed networks has received considerable attention in recent years. Although a large number of algorithms have been proposed for this problem, most of them are not proper for monster networks because of low performance, higher time complexity, and not being parameter-free. In the present investigation, the authors have proposed a new algorithm by employing the diffusion algorithm, which is in the category of diffusion methods. In the proposed algorithm, a weighted graph is developed, every single edge of which is a combination of the similarity of structures and attributes of two nodes that have an edge with each other. In the weighted graph, the influence of nodes will be calculated using Laplacian centrality. Afterward, the updating stage is performed, in which the node of a two-member set will have a label and label influence. Each node that supposes to be updated will select the label of node based on the higher influence of label among the adjacent nodes. Also, it will cause nodes with higher influence in terms of structure and attributes to update many tags. After a few steps, it is expected that the nodes which are homogeneous in terms of the structure of dense to have the same tags, which will be similar to tags of the same cluster of the graph. In the proposed algorithm, the efficiency of this method has been evaluated in comparison with other clustering methods of attributes, and it has been revealed that their high ratios are more effective in real and artificial datasets based on criteria such as modularity, NMI, and entropy. The proposed algorithm is linear in terms of time complexity and is superior to many algorithms in terms of time complexity; thus, it is suitable for large datasets. Based on the similarity criterion match coefficient for attributes, the proposed algorithm in the present study is appropriate for datasets which possess binary attributes; however, there are many datasets with multivalued or continuous attributes. Presenting a similarity criterion for such attributes that are useful for the diffusion algorithm can be considered as future research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alinezhad, E., Teimourpour, B., Sepehri, M.M., Kargari, M., 2020. Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches. Neural Comput. Appl. 32 (8), 3203–3220.

Amiri, S.E., Chen, L., Prakash, B.A., 2018. Efficiently summarizing attributed diffusion networks. Data Min. Knowl. Disc. 32 (5), 1251–1274.

Berahmand, K., Bouyer, A., 2018. LP-LPA: a link influence-based label propagation algorithm for discovering community structures in networks. Int. J. Mod. Phys. B 32 (06), 1850062.

Berahmand, K., Bouyer, A., 2019. A link-based similarity for improving community detection based on label propagation algorithm. J. Syst. Sci. Complexity 32 (3), 737–758.

Berahmand, K., Bouyer, A., Samadi, N., 2018a. A new centrality measure based on the negative and positive effects of clustering coefficient for identifying influential spreaders in complex networks. Chaos Solitons Fractals 110, 41–54.

Berahmand, K., Bouyer, A., Samadi, N., 2019. A new local and multidimensional ranking measure to detect spreaders in social networks. Computing 101 (11), 1711–1733.

Berahmand, K., Bouyer, A., Vasighi, M., 2018b. Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. IEEE Trans. Comput. Soc. Syst. 5 (4), 1021–1033.

Berahmand, K., Samadi, N., Sheikholeslami, S.M., 2018c. Effect of rich-club on diffusion in complex networks. Int. J. Mod. Phys. B 32 (12), 1850142.

Bothorel, C., Cruz, J.D., Magnani, M., Micenkova, B., 2015. Clustering attributed graphs: models, measures and methods. Netw. Sci. 3 (3), 408–444.

Chai, B.-F., Yu, J., Jia, C.-Y., Yang, T.-B., Jiang, Y.-W., 2013. Combining a popularity-productivity stochastic block model with a discriminative-content model for general structure detection. Phys. Rev. E 88, (1) 012807.

Chunaev, P. (2019). "Community detection in node-attributed social networks: a survey." arXiv preprint arXiv:1912.09816.

Chunaev, P., I. Nuzhdenko and K. Bochenina (2019). Community Detection in Attributed Social Networks: A Unified Weight-Based Model and Its Regimes. 2019 International Conference on Data Mining Workshops (ICDMW), IEEE.

Clauset, A., Newman, M.E., Moore, C., 2004. Finding community structure in very large networks. Phys. Rev. E 70, (6) 066111.

Faizal, E., 2014. Case based reasoning diagnosis penyakit cardiovascular dengan metode simple matching coefficient similarity. J. Teknol. Inform. Ilmu Komput. 1 (2), 83–90.

Fang, Y., Cheng, R., Luo, S., Hu, J., 2016. Effective community search for large attributed graphs. Proc. VLDB Endow. 9 (12), 1233–1244.

Garza, S.E., Schaeffer, S.E., 2019. Community detection with the label propagation algorithm: a survey. Phys. A. 122058.

Gibson, H. and J. Faith (2011). Node-attribute graph layout for small-world networks. 2011 15th International Conference on Information Visualisation, IEEE.

Greene, D. and P. Cunningham (2013). Producing a unified graph representation from multiple social network views. Proceedings of the 5th annual ACM web science conference.

Günnemann, S., Boden, B., Seidl, T., 2011. DB-CSC: a density-based approach for subspace clustering in graphs with feature vectors. Springer.

Gunnemann, S., I. Farber, B. Boden and T. Seidl (2010). Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. 2010 IEEE International Conference on Data Mining, IEEE.

Günnemann, S., I. Färber, S. Raubach and T. Seidl (2013). Spectral subspace clustering for graphs with feature vectors. 2013 IEEE 13th International Conference on Data Mining, IEEE.

Haghani, S., Keyvanpour, M.R., 2019. A systemic analysis of link prediction in social network. Artif. Intell. Rev. 52 (3), 1961–1995.

Huang, X., Cheng, H., Yu, J.X., 2015. Dense community detection in multi-valued attributed networks. Inf. Sci. 314, 77–99.

Karimi-Majd, A.M., Fathian, M., 2017. Multiobjective approach for detecting communities in heterogeneous networks. Comput. Intell. 33 (4), 980–1004.

Lancichinetti, A., Fortunato, S., Radicchi, F., 2008. Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78, (4) 046110.

Li, Y., Jia, C., Kong, X., Yang, L., Yu, J., 2017. Locally weighted fusion of structural and attribute information in graph clustering. IEEE Trans. Cybern. 49 (1), 247–260.

Liu, W., Lü, L., 2010. Link prediction based on local random walk. EPL (Europhys. Lett.) 89 (5), 58007.

Mohammadi, M., Moradi, P., Jalili, M., 2019. SCE: Subspace-based core expansion method for community detection in complex networks. Phys. A 527, 121084.

Moser, F., R. Colak, A. Rafiey and M. Ester (2009). Mining cohesive patterns from graphs with feature vectors. Proceedings of the 2009 SIAM International Conference on Data Mining, SIAM.

Perozzi, B., L. Akoglu, P. Iglesias Sánchez and E. Müller (2014). Focused clustering and outlier detection in large attributed graphs. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.

Pons, P., Latapy, M., 2005. Computing communities in large networks using random walks. International Symposium on Computer and Information Sciences. Springer.

Qi, X., Fuller, E., Wu, Q., Wu, Y., Zhang, C.-Q., 2012. Laplacian centrality: a new centrality measure for weighted networks. Inf. Sci. 194, 240–253.

Raghavan, U.N., Albert, R., Kumara, S., 2007. Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E 76, (3) 036106.

Rostami, M., Forouzandeh, S., Berahmand, K., Soltani, M., 2020. Integration of multi-objective PSO based feature selection and node centrality for medical datasets. Genomics.

Rosvall, M., Bergstrom, C.T., 2008. Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. 105 (4), 1118–1123.

Ruan, Y., D. Fuhry and S. Parthasarathy (2013). Efficient community detection in large networks using content and links. Proceedings of the 22nd international conference on World Wide Web.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T., 2008. Collective classification in network data. AI Mag. 29 (3), 93.

Steinhaeuser, K., Chawla, N.V., 2010. Identifying and evaluating community structure in complex networks. Pattern Recogn. Lett. 31 (5), 413–421.

Sun, H., Liu, J., Huang, J., Wang, G., Yang, Z., Song, Q., Jia, X., 2015. CenLP: A centrality-based label propagation algorithm for community detection in networks. Phys. A 436, 767–780.

vanDongen, S., 2000. A cluster algorithm for graphs. Inform. Syst. [INS] (R 0010).

Xu, Z., Y. Ke, Y. Wang, H. Cheng and J. Cheng (2012). A model-based approach to attributed graph clustering. Proceedings of the 2012 ACM SIGMOD international conference on management of data.

Yang, J., J. McAuley and J. Leskovec (2013). Community detection in networks with node attributes. 2013 IEEE 13th International Conference on Data Mining, IEEE.

Yang, T., R. Jin, Y. Chi and S. Zhu (2009). Combining link and content for community detection: a discriminative approach. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

Zarandi, F.D., Rafsanjani, M.K., 2018. Community detection in complex networks using structural similarity. Phys. A 503, 882–891.

Zhang, W., Zhang, R., Shang, R., Jiao, L., 2018. Weighted compactness function based label propagation algorithm for community detection. Phys. A 492, 767–780.

Zhou, Y., Cheng, H., Yu, J.X., 2009. Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. 2 (1), 718–729.

Zhou, Y., H. Cheng and J. X. Yu (2010). Clustering large attributed graphs: An efficient incremental approach. 2010 IEEE International Conference on Data Mining, IEEE.

Zhou, Y. and L. Liu (2013). Social influence based clustering of heterogeneous information networks. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.