Community Detection with Topological Structure and Attributes in Information Networks

ZHONGGANG WU, East China Normal University
ZHAO LU, East China Normal University & University of Pittsburgh
SHAN-YUAN HO, Massachusetts Institute of Technology

Information networks contain objects connected by multiple links and described by rich attributes. Detecting community for these networks is a challenging research problem, because there is a scarcity of effective approaches that balance the features of the network structure and the characteristics of the nodes. Some methods detect communities by considering topological structures while ignoring the contributions of attributes. Other methods have considered both topological structure and attributes but pay a high price in time complexity. We establish a new community detection algorithm which explores both topological Structure and Attributes using Global structure and Local neighborhood features (SAGL) which also has low computational complexity. The first step of SAGL evaluates the global importance of every node and calculates the similarity of each node pair by combining edge strength and node attribute similarity. The second step of SAGL uses a clustering algorithm that identifies communities by measuring the similarity of two nodes, calculated by the distance of their neighbors. Experimental results on three real-world datasets show the effectiveness of SAGL, particularly its fast convergence compared to current state-of-the-art attributed graph clustering methods.

CCS Concepts: • **Theory of computation → Unsupervised learning and clustering**; • **Computing methodologies → Cluster analysis**

Additional Key Words and Phrases: Community detection, global importance, topological structure, attribute similarity

ACM Reference Format:
Zhonggang Wu, Zhao Lu, and Shan-Yuan Ho. 2016. Community detection with topological structure and attributes in information networks. ACM Trans. Intell. Syst. Technol. 8, 2, Article 33 (November 2016), 17 pages.
DOI: http://dx.doi.org/10.1145/2979681

1. INTRODUCTION

Recently, the study of rich attribute data available for objects in real-world information networks has given rise to attributed graphs, where every vertex is characterized by a number of attributes describing the properties of the node, and the edges correspond to a topological structure [Papadopoulos et al. 2013]. As an expressive data

This work was supported by the Science and Technology Commission of Shanghai Municipality, under grant 14511107000, grant 16511102702, and grant 14DZ2260800.
Authors' addresses: Z. Wu, Department of Computer Science and Technology, East China Normal University, No. 3663 North Zhongshan Road, 200062 China; email: wallacewu1203@gmail.com; Z. Lu (corresponding author), Department of Computer Science and Technology, Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, No. 3663 North Zhongshan Road, 200062 China; email: zlu@cs.ecnu.edu.cn; S.-Y. Ho (corresponding author), Department of Electrical Engineering & Computer Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Room 36-541, Cambridge, MA 02139, USA; email: hoho@mit.edu.

DOI: http://dx.doi.org/10.1145/2979681

structure, an attributed graph is used in many application domains, for example, web link networks, social media, and coauthor networks. The task of community detection in information networks refers to clustering nodes of attributed graphs into communities, where a community (or a cluster) means a group of nodes that are inclined to flock together [Dang and Viennet 2012]. Clustering attributed graphs has led to a number of approaches that can be classified into three categories: structure based, attribute based, or structure and attributes based [Zhou et al. 2010]. Newman partitions information networks by minimizing interconnection to detect community [Newman 2006, 2013]. Two approaches [Chen and Saad 2012; Singh and Awekar 2013] identify clusters with high structure density, and other methods [Dev 2014; Weng et al. 2014] detect communities using hierarchical clustering algorithms. One main issue of the aforementioned approaches is that they only focus on tightness of links (or edges) but ignore attributes associated with nodes. The Summarization by grouping Nodes on Attributes and Pairwise relationships (SNAP) algorithm and the k-SNAP algorithm and k-SNAP [Tian et al. 2008] group nodes with similar attributes, but ignore links among nodes. These links reveal the important fact that connected nodes may belong to the same community even though their attribute values are diverse.

Characterizing both topological structures and node attributes simultaneously is a challenging task because of the diverse characteristics of structures and attributes. The Structure-Attribute Clustering (SAC1) model calculates increments of modularity similarities and attribute similarities through utilizing a weight factor [Dang and Viennet 2012]. Other methods are mostly based on a probabilistic model [Xu et al. 2012; Yang et al. 2013]. The authors Zhou et al. [2009, 2010] and Cheng et al. [2011] suggested the Structural and Attribute graph Clustering (SA-Cluster) approach and its extensions, a method to calculate the distance between two nodes and its extensions using a unified random-walk algorithm. However, these distance-based random-walk approaches have higher time complexity (e.g., SA-Cluster is $O(n^3)$, where n is the number of nodes), and none of these approaches analyzed the global importance of nodes which represents the topological structure of the network.

From a technical perspective, the goal of community detection of an attributed graph is to obtain clusters that are not only densely connected but also homogeneous. The challenges are as follows. First, real networks are extremely sparse, that is, nodes connect to only a very small fraction of nodes in the network, and thus, sole global analysis over a community structure can result in unbalanced scales of communities due to the simplified assumption that the entire network exhibits a uniform number of connections and attributes at each node [Aggarwal et al. 2011]. The second challenge of characterizing nodes and their interactions is more subtle: that of characterizing their various characteristics.

We observe that each node links to other nodes, all connected nodes can be viewed as a neighborhood, and all nodes belonging to the same neighborhood have common characteristics. From this perspective, we compare the similarity of two nodes by computing similarities between their neighbors. Specifically, our contributions are summarized below.

(1) We developed the SAGL algorithm, an efficient method that combines the topological structure and node attributes characteristics based on analyzing global structure and node neighbors. We adopted the PageRank algorithm to analyze the global structure to obtain node importance. We then analyzed edge strength and node attribute similarity and combine them by a weight factor.
(2) We designed a new algorithm to measure similarities among nodes. During the clustering procedure, we compute the similarity between the centroid of a cluster and the new node that will be assigned to this cluster, according to neighbor similarity.

Table I. Some Common Used Community Detection Approaches for Attributed Graphs

Algorithm	Global structure	Local structure	Attribute weights	Multi-graph
CONCLUDE [De Meo et al. 2014]	√	√		
HASCOP [Papadopoulos et al. 2013]		√	√	√
BAGC [Xu et al. 2012]		√	√	
k-SNAP [Tian et al. 2008]		√	√	
SA-Cluster [Cheng et al. 2011]		√	√	
SAGL	√	√	√	

(3) The proposed method is performed on two real-world networks. Experimental results demonstrate that the task of detecting latent communities for an attributed graph is both effective and efficient.

This article is organized as follows. Section 2 reviews related works. Section 3 describes our problem. Section 4 details our work on combining topological structure and node attributes. Section 5 proposes a clustering algorithm of grouping nodes into clusters. Experimental results are contained in Section 6. Finally, we conclude our work in Section 7.

2. RELATED WORK

Community detection in attributed networks is still a challenging research topic that has attracted diverse data resources, along with a number of different techniques implemented. Attributed graph clustering algorithms use a similarity or distance measure that combines both structural and attribute information of the nodes and then implements a variety of graph clustering algorithms. These algorithms can be categorized into three types: structure based, attribute based, or structure and attribute based. The features of some typical algorithms are summarized in Table I.

2.1. Structural Information Analysis

There are two types of structural information, global and local, in an attributed graph. Global structural analysis studies nodes or edges in the entire network as a whole, and their methods exploit knowledge about the network topology to find clusters. Local structural analysis is concerned with partial knowledge of the network topology, e.g., the neighbors of a vertex.

Global Structural Analysis. Girvan and Newman [2002] observed that the property of community structure, which says network nodes are joined together in tightly knit groups in a community, while connections between communities are looser. They used a method for detecting communities with the above properties by extending *betweennes centrality* to count the number of shortest paths that pass by a given edge in a network.

Ding et al. [2009] used a *PageRank* algorithm to rank authors in co-citation networks. They discussed how varied damping factors in the PageRank algorithm can provide additional insight into the ranking of authors in an author co-citation network. Based on their research, they further proposed weighted PageRank algorithms. Their experiments on datasets using damping factors ranging from 0.05 to 0.95, show that citation rank is highly correlated with PageRank, with the only difference in the damping factors.

Global approaches are able to achieve high values of modularity, but they do not scale well on large networks, and thus, cannot be applied to the analysis of the attributed graph. In our study, the PageRank algorithm is used to measure the global structural information of a vertex in an attributed graph.

Local Neighbors Analysis. Zhou et al. [2009] (refers to Zhou and Lü et al. [2009]) reported that *common neighbors* have the best overall performance. They further investigated a variety of local structure similarity measures, such as cosine similarity, common neighbors, and Jaccard similarity.

Singh and Awekar [2013] proposed a clustering method, the Incremental Shared Nearest Neighbor Density-based clustering (IncSNN-DBSCAN) method, based on incremental shared nearest neighbor to extract density regions. Their method is an order-independent, incremental version of the Shared Nearest Neighbor Density-based clustering (SNN-DBSCAN) algorithm. However, a bottleneck of the algorithm is the need for significant memory overhead.

The COmplex Network CLUster DEtection (CONCLUDE) algorithm has two stages and uses information at the global level [De Meo et al. 2014]. It first computes the importance of each edge in keeping the network connected (i.e., edge centrality) by using an information propagation model based on random and non-backtracking walks of finite length. Then, it uses edge centrality to map network vertices onto points of an Euclidean space to compute distances between all pairs of connected vertices. The second stage uses the distances computed in the first stage to partition the network into clusters.

In this study, we combine both global and local structural information to measure similarities among nodes and clusters.

2.2. Attribute Information Analysis

The attributed-based algorithms aim to partition the graph according to attribute similarity, so vertices with the same attribute values are grouped into one partition.

The SNAP and k-SNAP algorithms are used to summarize graphs [Tian et al. 2008]. Specifically, SNAP produces a summary graph by grouping nodes based on user-selected node attributes and relationships, while k-SNAP further allows users to control the resolutions of summaries. These two algorithms group nodes with similar attributes while ignoring links among nodes. However, links reveal the fact that connected nodes may belong to the same community even though their attribute values might possibly be quite different.

Some other algorithms use probabilistic models, for example, Bayesian or expectation-maximization (EM), to cluster attributed graphs. Generally, a cluster is modelled by its connection and attribute distributions. These approaches calculate the parameters of the distributions, and a vertex is categorized into a cluster if its properties follow the cluster's distributions. The Bayesian Attributed Graph Clustering (BAGC) approach uses Bayesian inference to cluster attributed graph by incorporating node attributes as augmented edges [Xu et al. 2012]. This method deals with only one type of link.

2.3. Structure and Attribute-Based Analysis

In many real life applications, the graph topological structure and node attributes are most important. Thus, methods must consider both structural and attribute information of an attributed graph.

SA-Cluster calculates the random-walk distance to evaluate whether a node has closeness on an augmented attributed graph [Zhou et al. 2009]. The original graph has added attribute nodes and edges in order to connect nodes with shared attribute

values. The weight of each edge depends on the importance of the attribute that the attribute vertex represents. By enriching the graph, vertices that share the same attributes are closer, and there exists a path of at least two hops between them, through the attribute vertex. Based on this distance measure, SA-Cluster takes a K-Medoids clustering algorithm to partition the graph. At each iteration, the attribute weights are recalculated and the random-walk distances on the augmented attributed graph are also recalculated. In contrast, our SAGL algorithm clusters nodes by measuring the similarities of the nodes' neighbors. The authors further provided some optimization techniques on matrix computation to improve the efficiency of SA-Cluster on large graphs [Cheng et al. 2011].

SAC1 and SAC2 also proposed a method to discover communities in an attributed graph [Dang and Viennet 2012]. SAC1 is based on the modification of Newman's well-known modularity function through defining the *modularity attribute* of a partition using *simA*, an attribute similarity function. To determine the attribute similarity between two communities, they used two approaches, summing up the similarity of their members and setting the similarity of their centroids. SAC1 calculated modularity increments, which combines structure modularity and attribute modularity by a weight factor. In the modularity increment techniques, a node is classified to a certain cluster or stays in its original state. To reduce the $O(n^2)$ complexity of SAC1, they proposed SAC2. This second method first constructs a simplified graph by using k-nearest neighbors and then partitions the graph by calculating modularity increments.

Papadopoulos et al. [2013] proposed the Homogeneity Attributes and Similar COnnectivity Patterns (HASCOP) algorithm for clustering attributed multi-graphs. The two main steps are the assignment of vertices into clusters, and the adjustment of the link type and attribute weights. The Communities from Edge Structure and Node Attributes (CESNA) model [Yang et al. 2013] detected *overlapping* or *non-overlapping* communities in a network using the network structure and the features and attributes of nodes. They use a probabilistic model which combines community memberships, the network topology, and node attributes.

Other work focused on community detection and visualization [Cruz et al. 2013], detecting structural and overlapping information in ground-truth communities [Yang and Leskovec 2014], tracking the evolution from a network stream [Tang and Yang 2014], employing the background knowledge contained in the descriptions of nodes in the communities, or tackling the influence maximization problem through detecting communities [Chen et al. 2014].

To summarize, existing methods all ignore the global importance of nodes in view of its topological structure in an attributed graph. Moreover, these methods add nodes, which lead to an explosion of the graph. Since the initial attributed graph typically has rich attributes and attribute values, memory and runtime requirements increase dramatically.

Our work bears the most similarity to SA-Cluster. The difference between SA-Cluster and SAGL are the following: first, SA-Cluster considers only the local structure of nodes in an attributed graph, and assigns the same weight to all topological links. In contrast, SAGL combines both global structure and local neighbors, by assigning different weights to different topological links. Second, before clustering, SA-Cluster calculates similarities of all node pairs through computing the distance based on a random walk. SAGL calculates similarities of node pairs by computing the similarities of local neighborhood nodes when assigning nodes to their relevant clusters. Thus, not all local similarities are computed for all node pairs. Third, SA-Cluster generates an augmented attributed graph by considering attributes as a new attribute node, which causes increased time complexity and space consumption. SAGL does not increase the

Table II. Notations Used in the Paper

V	Set of nodes in an attributed graph.
E	Set of edges in an attributed graph.
X	Set of node attributes.
K	Number of communities in an attributed graph.
λ	Weight factor for level of impact of topological structure and attributes.
σ	Influence parameter.
C_k	The kth partition (community).
$N_I(v_i)$	Set of in-neighbor nodes of node v_i.
$N_O(v_i)$	Set of out-neighbor nodes of node v_i.
$N_L(v_i)$	Set of local neighbors which contain in-neighbor and out-neighbor nodes of node v_i (includes v_i).
$D_F(v_i, v_j)$	Distance factor between v_i and v_j.
$A_{\lvert V\rvert\times\lvert V\rvert}$	Adjacent matrix of an attributed graph.
$T_C(v_i, v_j)$	Topological closeness between v_i and v_j.
$S_{NA}(v_i, v_j)$	Attribute Similarity between v_i and v_j.
$S_N(v_i, v_j)$	Node similarity that combines topological closeness and attribute similarity between v_i and v_j.
$S_{NB}(v_i, v_j)$	Neighbor Similarity between v_i and v_j.
$S_{avg}(\overline{v}_i, k)$	Similarity between "average point" $\overline{v}_i$ in cluster C_k.

size of the attributed graph, because it utilizes both structure and attribute information to detect clusters.

3. PROBLEM STATEMENT

Denote an *attributed graph* as $G = (V, E, X)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, $E \subset V \times V$ is the set of directed edges that connect nodes, and $X = \{a_1, a_2, ..., a_m\}$ denotes the set of attributes owned by the nodes. The values of all attributes of the node v_i are represented as $\{a_1(v_i), a_2(v_i), \ldots, a_m(v_i)\}$. The nodes that have a directed link to node v_i are viewed as *in-neighbors* of node v_i, denoted as $N_I(v_i)$. Similarly, the nodes that node v_i has directed links to are viewed as the *out-neighbors* of node v_i and represented as $N_O(v_i)$. Define the *local neighbors* $N_L(v_i)$ of node v_i as the union of in-neighbors, out-neighbors, and itself $N_L(v_i) = N_I(v_i) \cup N_O(v_i) \cup v_i$. For simplicity, we use the word *neighbors* to refer to local neighbors unless otherwise specified. The symbols are summarized in Table II.

The task of detecting communities in an attributed graph G is finding a partition $(C_1, C_2, \ldots, C_K)$, such that, for each C_k, $1 \leq k \leq K$, a community of graph G, is non-empty, the C_k's are disjoint, and each C_k is a connected graph. That is, $V = \bigcup_{k=1}^{K} C_k$ and $C_i \cap C_j = \emptyset$, for $i \neq j$.

A desired clustering of attributed graphs should achieve a good balance between the following desired qualities: (i) Nodes within one cluster are close to each other in terms of structure, while nodes between clusters are distant from each other, and (ii) nodes within one cluster have similar attribute values while nodes between clusters could have quite different attribute values.

Given the above definitions, we decompose the main task in community detection in an attributed graph into two sub-tasks: (1) devise a distance measure that evaluates topological closeness between two nodes from the characteristics of their neighborhood and determine similarities of nodes, and (2) construct a clustering algorithm that uses the most optimistic centroids initialization at each iteration.

The main steps of tackling the main issues above are summarized here: (1) Identify the *global importance* $g_i \in [0, 1]$ for each node v_i and calculate the *edge strength* $T(v_i, v_j)$ of each node pair v_i and v_j; (2) measure the *topological closeness* $T_C(v_i, v_j)$ of two nodes through measuring the neighbors $N_L(v_i)$ and $N_L(v_j)$ for two nodes v_i and v_j; and

(3) cluster all nodes into groups. These steps will be detailed and discussed in the following sections.

4. THE PROPOSED APPROACH

The basic ideas of our method are the following. We first compute the topological closeness of nodes by calculating their global importance and edge strengths. We next measured attribute similarities of nodes using various attribute values. Finally, we devised a balanced strategy to combine structure closeness and attribute similarity.

4.1. Measuring Topological Structure

The topological structure of an attributed graph G describes the connectedness among all nodes in the graph. We create an adjacent matrix A to describe the conductivities among nodes. If there is a directed edge from node v_i to node v_j, then the value of A_{ij} is 1, otherwise A_{ij} is 0. The ith row and the ith column in A denote the out-neighbors $N_O(v_i)$ and in-neighbors $N_I(v_i)$ of node v_i, respectively.

We use the link analysis algorithm PageRank [Arasu et al. 2001] to obtain the weights of each node using the topological structure of graph G. The numerical weights assigned to each node in the graph measures its relative importance. The global importance g_i of node v_i is defined by the PageLink value,

$$g_i = \frac{1-d}{|V|} + \sum_{v_k \in N_I(v_i)} \frac{g_k \times d}{|N_O(v_k)|}, \tag{1}$$

where the damping factor d represents the probability that a node continues to link to other nodes, and $|V|$ is the number of nodes in the network. The higher the value of d, the more important the node. Yan and Ding [2011] investigated varying values of the damping factor d from 0 to 1. They chose damping factors of 0.85, 0.75, 0.65, 0.55, 0.45, 0.35, 0.25, and 0.15, and experiments always showed that a damping factor of 0.85 demonstrated the highest effectiveness. Using their results, we also set d to 0.85 in our experiments.

We use $T_e(v_i, v_j)$ to describe the *edge strength* from node v_i to node v_j and vice versa. For (i, j), $T_e(v_j, v_i)$, $T_e(v_i, v_j) \in [0, 1]$ is computed from Equation (2),

$$T_e(v_i, v_j) = \begin{cases} \frac{1}{1+exp\left(-\frac{g_i}{|N_O(v_i)|} \cdot g_j\right)}, & \text{if } A_{ij} = 1. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We employed the exponential function in Equation (2) to keep the range of attribute similarity close to the range of structure similarity. Equation (2) says that if v_j has higher importance than the nodes belonging to the out-neighbors $N_O(v_i)$ of node v_i, the edge strength from node v_i to node v_j will be stronger. The *topological closeness* $T_C(v_i, v_j)$ between two nodes v_i to v_j is the sum of all edge strengths from v_i to v_j and vice versa. Thus,

$$T_C(v_i, v_j) = T_e(v_i, v_j) + T_e(v_j, v_i). \tag{3}$$

4.2. Measuring Attributes of Nodes

In addition to the topological structure of nodes, attributes of nodes also play an important role in detecting homogeneous clusters. We employed various strategies to measure attribute similarities among nodes according to the value types of attributes.

If an attribute has a numerical value, then we can compare and assign the value of *attribute dissimilarity* $\zeta_{a_{ijr}}$ to 1 if the value of attribute a_r of v_i and v_j differs. Otherwise, $\zeta_{a_{ijr}}$ is assigned to 0. If the value of an attribute is text, then we adopt the Latent Dirichlet Allocation (LDA) algorithm [Blei et al. 2003] to analyze topic distribution.

Then each text is assigned to a topic that has the highest probability. The variable $\zeta_{a_{ijr}}$ measures whether two nodes have the same topic. If they are the same, then it is set to 0, otherwise it is set to 1.

Supposing that a node v_i has an attribute vector $\{a_1(v_i), a_2(v_i), \ldots, a_m(v_i)\}$ and corresponding weight vector $\{w_{a_1}, w_{a_2}, \ldots, w_{a_m}\}$, then the *attributes similarity* $S_{NA}(v_i, v_j)$ is computed with the following weighted Euclidean distance:

$$S_{NA}(v_i, v_j) = \frac{1}{1 + \sqrt{\sum_{r=1}^{m} w_{a_r} \cdot \zeta_{a_{ijr}}^2}}. \tag{4}$$

Here, the value of $S_{NA}(v_i, v_j)$ is limited in (0,1]. At each iteration, the weights w_{a_r} are adjusted by a voting mechanism that automatically adjusts the weights of attributes suggested in Zhou et al. [2009].

4.3. Combining Topological Structure and Node Attributes

We use a *weight factor* λ to evaluate the node similarity $S_N(v_i, v_j)$ for two nodes v_i and v_j, expressed by

$$S_N(v_i, v_j) = (1 - \lambda) \cdot T_C(v_i, v_j) + \lambda \cdot S_{NA}(v_i, v_j), \tag{5}$$

where $\lambda \in [0, 1]$. Note that if $\lambda = 0$, then the characteristics of node attributes are ignored. At the other extreme, if $\lambda = 1$, then the features of topological structure are not considered. The aim is to balance the influence of topological structure closeness and node attributes similarity.

Given two nodes v_i and v_j, their *neighbor similarity* $S_{NB}(v_i, v_j)$ is measured by

$$S_{NB}(v_i, v_j) = \sum_{v_i' \in N_L(v_i)} \sum_{v_j' \in N_L(v_j)} \frac{S_N(v_i', v_j') \cdot D_F(v_i', v_j')}{|N_L(v_i)| \cdot |N_L(v_j)|}, \tag{6}$$

where $D_F(v_i', v_j')$ is a *distance factor* that measures the influence between two nodes v_i' and v_j'. Similar to Yu et al. [2015], we define this factor to be

$$D_F(v_i', v_j') = e^{-(\frac{d(v_i', v_j') - 1}{\sigma})^2}.$$

Here, σ is the *influence parameter*. The larger the value of σ, the more influence of neighbor-based similarity on the similarity measurement of two nodes. We set the same value for σ to the control influence parameter for the experimental datasets.

Our observation shows that the contributions of neighbors vary according to their types. In this study, we define two types of nodes: the node itself and its 1-step neighbor. We define the three values for $d(v_i', v_j')$ as the following:

$$d(v_i', v_j') = \begin{cases} 1, & \text{if } v_i' \text{ is } v_i, \text{ and } v_j' \text{ is } v_j. \\ 2, & \text{if } v_i' \text{ is a 1-step neighbor of } v_i \text{ and } v_j' \text{ is } v_j, \\ & \text{or } v_i' \text{ is } v_i \text{ and } v_j' \text{ is a 1-step neighbor of } v_j. \\ 3, & \text{if } v_i', v_j' \text{ are 1-step neighbors of } v_i \text{ and } v_j. \end{cases} \tag{7}$$

5. CLUSTERING ALGORITHM

The second component of our method is using a clustering algorithm to partition the attributed graph G according to both topological structure and node attributes. We employed the *K-Medoids* algorithm [Kaufman and Rousseeuw 1987] where the global importance of a node indicates its centrality in the graph, and the node with the highest global importance will be selected as an initialization of a cluster centroid. During each iteration, the most centrally located nodes are selected as cluster centroids. Other nodes

are assigned to their closest clusters by calculating their neighbor similarity and the centroid of each cluster.

5.1. Centroid Initialization

One key component of the clustering algorithm K-Medoids is selecting optimal initial centroids [Zhou et al. 2009]. The algorithm sets the initial centroids of all clusters randomly. If the algorithm is used directly, then there are two challenges. First, the importance of nodes in the attributed graph will be neglected, which will cause many iterations and a lower convergence speed. Second, if a node with a few neighbors is selected as an initialized centroid, few nodes will be assigned to its cluster, and thus the sizes of the detected clusters will be extremely unbalanced. To tackle these issues, we choose the top K nodes according to their neighbor similarity as initial centroids of clusters.

5.2. Centroid Updating

Another important issue of the clustering process is the strategy of updating the centroids of clusters. We assume that a node can be affected by their neighbors. For example, in a citation network, a article can cite other articles (out-neighbors) or be cited (in-neighbors). To some extent, these cited and citing articles have a similar research topic. Thus, we select the following strategy. During the clustering process, whether a node v_i is assigned to a cluster C_k or not is determined by the closeness of the node v_i with the centroid in the cluster. In the nth iteration, the node v_i is assigned to the cluster C_k if the neighbors similarity $S_{NB}(v_i, c_k^n)$ between the node v_i and the centroid c_k^n is the maximum. That is,

$$\arg\max_{c_k^n} S_{NB}\left(v_i, c_k^n\right). \tag{8}$$

This algorithm is used to assign all nodes to their new clusters. To update the centroid of each cluster, we first compute the similarity between the "average point" $\overline{v}_i$ and the cluster C_k by

$$S_{avg}\left(\overline{v}_i, k\right) = \frac{1}{|C_k|} \sum_{v_i', v_j' \in C_k} S_N\left(v_i', v_j'\right). \tag{9}$$

For the $(n+1)$-th iteration, node $v_{i'}$ is selected as a new centroid c_k^{n+1} if there exits $v_{i'}$,

$$\begin{aligned} \arg\min_{v_i' \in C_k} \left| \sum_{v_j' \in C_k} S_N\left(v_i', v_j'\right) - S_{avg}\left(\overline{v}_i, k\right) \right|, \\ s.t. \quad |N_L\left(v_i'\right)| \geq D_{avg}\left(C_k\right). \end{aligned} \tag{10}$$

We choose a centroid that is not only centrally located but also has more connectivity to other nodes in a cluster. We choose the node with a higher degree than the *average degree* $D_{avg}(C_k)$ of cluster C_k, that is, the average number of neighbors, as shown in Equation (11),

$$D_{avg}\left(C_k\right) = \frac{\sum_{v_i' \in C_k} |N_L\left(v_i'\right)|}{|C_k|}. \tag{11}$$

At the conclusion of all the steps above, a node is selected for a cluster if its neighbors are higher than that of all other nodes in the cluster.

ALGORITHM 1: The SAGL Algorithm

Input: Attriubted graph $G(V, E, X)$, Community number K, weight factor λ.
Output: K communities.
$w_{a_i} = 1$;
Calculate node global importance b_i;
Measure topological closeness $T_C(v_i, v_j) = S_E(v_i, v_j) + S_E(v_j, v_i)$;
Calculate attribute similarity $S_A(v_i, v_j)$ using Equation (4);
Calculate $S_N(v_i, v_j) = (1 - \lambda) \cdot T_C(v_i, v_j) + \lambda \cdot S_A(v_i, v_j)$;
Initialize K clusters centroids;
repeat
  Assign v_i to Cluster C_k with a centroid c' according Equation (8);
  Update cluster centroid c_k according to using Equation (10);
  Do Attribute weight self-adjustment;
  Re-calculate $S_N(v_i, v_j)$ using Equation (5);
until *Objective function converges*;

5.3. Object Function

To determine that all nodes of clusters are more similar than that of all nodes among clusters, we first calculate the *intra-cluster similarity* $S_{clu}(C_k)$ for a cluster C_k by computing similarities of all nodes within the cluster as

$$S_{clu}(C_k) = \sum_{v_i, v_j \in C_k} S_N(v_i, v_j).$$

The objective is to compute the largest similarities of these clusters by maximizing the *objective function Obj*,

$$Obj\{C_k\}_{k=1}^{K} = \sum_{k=1}^{K} \frac{S_{clu}(C_k)}{|C_k| \cdot |C_k|}. \tag{12}$$

The objective function *Obj* considers not only the node similarities within a cluster but also the scales of clusters. We employ the objective function to avoid the following unreasonable partition. If clusters of a partition are extremely unbalanced in size, for example, the objective function value will be lower. The higher the value of the objective function, the better the graph partition. Our approach is summarized in Algorithm 1.

For our experiments, we ran the algorithm for λ ranging from 0.1 to 0.18 with step size 0.02. SAGL obtains the best clustering quality when $\lambda = 0.14$ for the Political Blogs Dataset and $\lambda = 0.12$ for the DBLP-5000 Dataset.

If we only consider topological structure and neglect the contributions of nodes' attributions in SAGL (if we set $\lambda = 0$) in Equation (5), then we will get the Topological Structure with Global and Local information (TSGL) approach. That means TSGL, a topological structure-based method, is a special case of SAGL.

6. EXPERIMENTS AND ANALYSIS

SAGL has been implemented in *Java 1.8 on commodity hardware with 2.5GHz Central Processing Unit (CPU) and 8GB main memory.* To evaluate the proposed approach, SAGL and its special case TSGL, as discussed at the end of Section 5, are compared to the commonly used algorithms k-SNAP [Tian et al. 2008], SA-Cluster [Zhou et al. 2009], and BAGC [Xu et al. 2012].

Table III. Datasets

Dataset	$\|V\|$	$\|E\|$	Attributes	Link types
DBLP-5K	5,000	35,740	2	1
DBLP-10K	10,000	55,734	2	1
Political Blogs	1,490	19,087	1	1

6.1. Experimental Datasets

Table III shows the properties of the datasets, which includes the number of vertices, edges, attributes, and link types of three information networks used in our experiments. Note that the edges that connect vertices are ignored.

(1) DBLP Bibliograph dataset (DBLP-5000): The DBLP-5000 dataset is generated in Zhou et al. [2009]. The authors build a coauthor graph with the top 5,000 authors and their coauthor relationships. They use two relevant attributies: *prolific* and *primary topic*. For attribute "prolific," authors with $\geq$20 articles are labeled as highly prolific; authors with $\geq$10 and $\leq$20 articles are labeled as prolific, and authors with $<$10 articles are labeled as low prolific. For attribute "primary topic," they used a topic modeling approach to extract 100 research topics from a document collection composed of article titles from the selected authors. Each extracted topic consists of a probability distribution of keywords that are most representative of the topic. Each author has 1 of 100 topics as his/her primary topic.

(2) DBLP Bibliograph dataset (DBLP-10K): This dataset is a subset of the complete DBLP dataset and contains 10K vertices that represent the top authors from the complete DBLP dataset [Xu et al. 2012]. Each vertex represents an author described by two attributes: "prolific" and "primary topic." There are four topics, which are databases (DB), data mining (DM), information retrieval (IR), and artificial intelligence (AI). An edge (v_i, v_j) represents that authors with ID's i and j have co-authored at least one publication.

(3) Political Blogs dataset (Political Blogs): The third dataset is a directed network of 1,490 weblogs with 19,090 hyperlinks on US politics recorded in 2005 by Adamic and Glance [2005]. Each weblog has an attribute describing its political leaning as either *liberal* or *conservative*.

6.2. Comparison Methods and Evaluation

In this section, comparison experiments for SAGL and its variant (TSGL), along with two other algorithms, SA-Cluster and k-SNAP, are shown.

We use the following *Density* and *Entropy* functions to evaluate the quality of the detected communities [Zhou et al. 2009] in Equations (13) and (14),

$$Density\{C_k\}_1^K = \sum_{k=1}^{K} \frac{\{(v_x, v_y)|v_x, v_y \in C_k, (v_x, v_y) \in E\}}{|E|}, \tag{13}$$

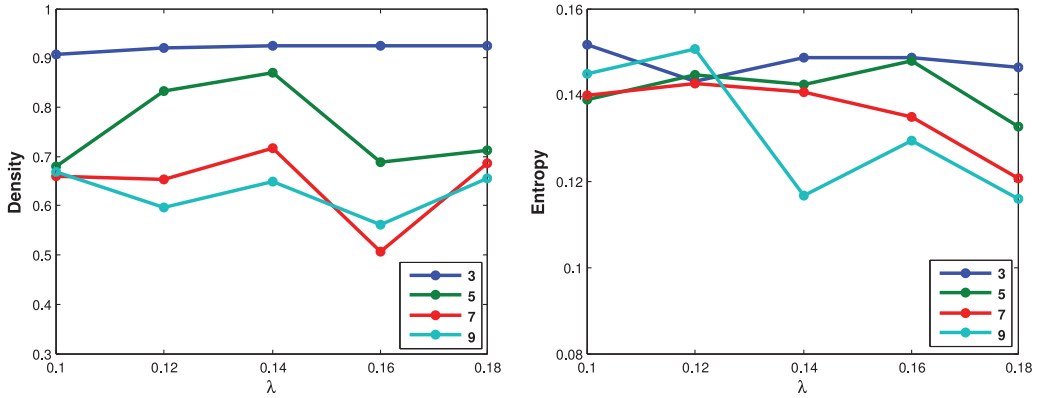
$$Entropy\{C_k\}_1^K = \sum_{i=1}^{m} \frac{w_i}{\sum_{r=1}^{m} w_r} \sum_{k=1}^{K} \frac{|C_k|}{|V|} entropy(a_i, C_k), \tag{14}$$

where

$$entropy(a_i, C_k) = -\sum_{\gamma=1}^{T_i} p_{ij\gamma} log_2 p_{ij\gamma}.$$

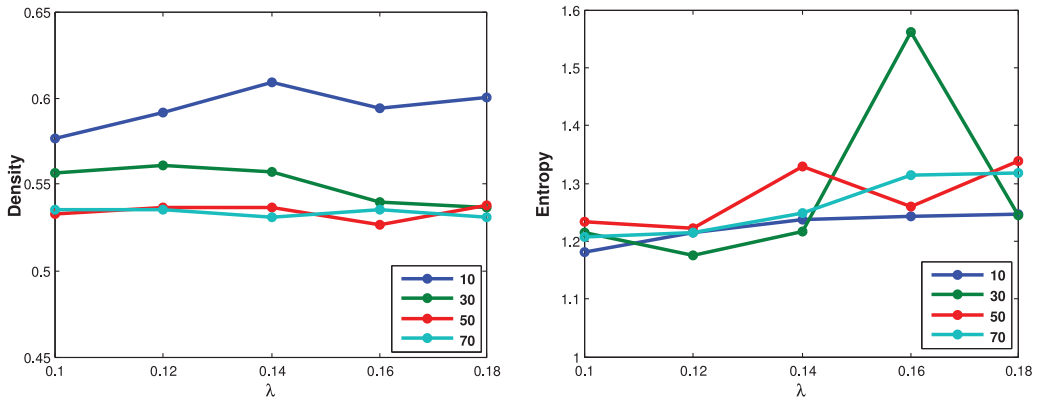
The entropy for each attribute is defined as

$$entropy(a_i) = \frac{|C_k|}{|V|} entropy(a_i, C_k).$$



(a) Density (b) Entropy

Fig. 1. Effect of varying λ on Political Blogs. The curves in blue, green, red, and light blue represent the experimental results when $K = 3, 5, 7,$ and 9, respectively.



(a) Density (b) Entropy

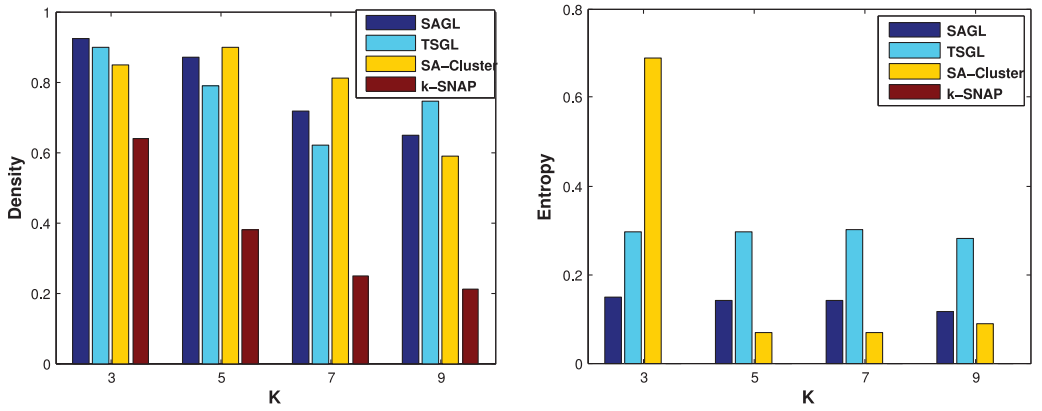
Fig. 2. Effect of varying λ on DBLP-5000. The curves in blue, green, red, and light blue represent the experimental results when $K = 10, 30, 50,$ and 70, respectively.

Here, $p_{ij\gamma}$ is the percentage of nodes within the cluster C_k that has values $a_{it} \in \{a_{i1}, a_{i2}, \ldots, a_{iT_i}\}$ for the attribute a_i. Note that T_i is the number of variety values obtained by attribute a_i. Communities with low entropy mean they are more homogeneous.

6.3. Measuring Weight Factor

Our approach used a weight factor λ to adjust the impact of topological structure and attributes. Different values of the weight factor lead to diverse community quality. We first conduct experiments to detect the relationships between community quality and the weight factor using SAGL. We show the curves of the density and entropy of the detected communities with varied λ, both on Political Blogs and DBLP-5000 in Figures 1 and 2, respectively.

In the two experiments, the influence parameter is set to $\sigma = 3.5$. From Figure 1, we observe that the best quality of clustering on the Political Blogs dataset is obtained when $\lambda = 0.14$. According to the evaluation metrics in Section 6.2, a good clustering result will be one with the characteristics of higher density and lower entropy. Figure 1



(a) Density (b) Entropy

Fig. 3. Cluster quality comparison of current state-of-the-art methods on Political Blogs.

shows the effect of varied λ on Political Blogs. The density curves increases when $\lambda \leq 0.14$ and decreases when $\lambda > 0.14$. In Figure 1(b), the entropy curve of the detected clusters have mostly a downward trend. Based on the above observations, we see that $\lambda = 0.14$ is the best value when density and entropy are synthetically considered.

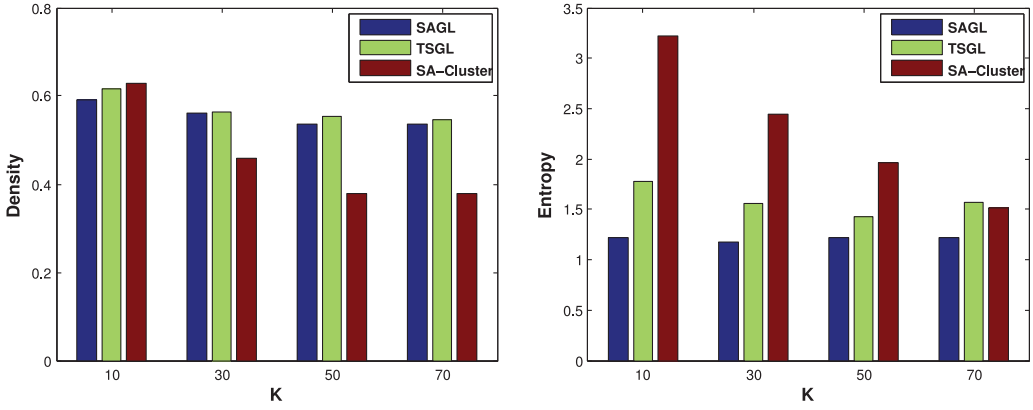
Figure 2 shows experimental results on the DBLP-5000 dataset. Following the same evaluation criterion, the best clustering quality occurs when $\lambda = 0.12$. From Figure 2, the change in values of both density and entropy are extremely slight. Our investigation shows that the reason is that DBLP-5000 has two attributes, *primary topic* with 100 topics and *prolific* with three values (i.e., high, medium, and low). These lead to more choices of attribute values and avoid over-concentration of nodes into several clusters. When λ increases, node attributes become more important in the clustering procedure and lead to more homogeneous clusters.

6.4. Clustering Quality Evaluation

For further study of the detection of community quality, we compare our two methods, SAGL and TSGL, with other state-of-the-art methods in two aspects, community density and entropy. Experimental results of SA-Cluster and k-SNAP on DBLP-5000 and Political Blogs can be found in Zhou et al. [2009], while experimental results of BAGC and SA-Cluster on the DBLP-10K dataset can be found in Xu et al. [2012].

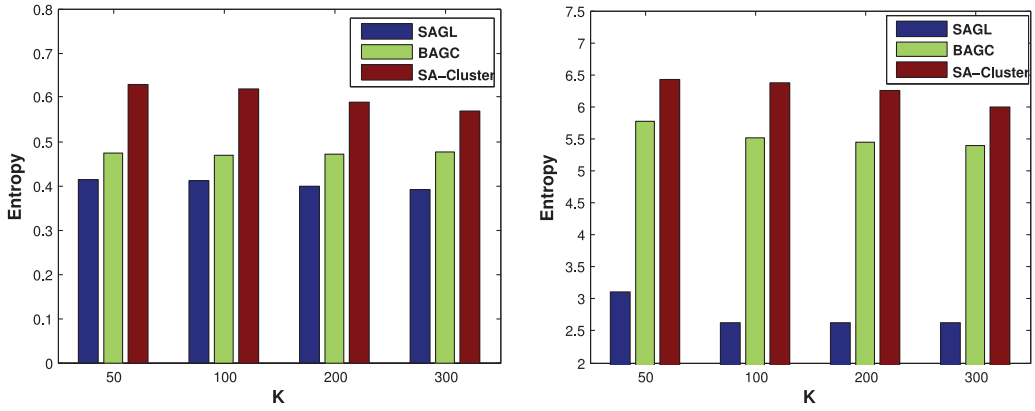
Clustering Results on Political Blogs. For the Political Blogs dataset, all approaches are executed with community number $K = 3, 5, 7, 9$ for clustering quality comparison. The comparison of density and entropy is shown in Figure 3(a). SAGL achieves a density of about 0.92 with the highest when $K = 3$. SAGL achieves a better result with higher density and lower entropy than that of TSGL due to the influences of node attributes. SA-Cluster has approximately the same density with SAGL and TSGL when K is increased. k-SNAP has the lowest density in the four methods and decreases sharply as K increases. The reason is that k-SNAP partitions a graph based only on node attributes without taking into account node connectivity.

Clustering entropy comparison with a different community number is shown in Figure 3(b). SAGL achieves the lowest entropy, about 0.15 when $K = 3$. SAGL, which utilizes topological structure and node attributes, leads to densely connected as well as homogeneous communities. SA-Cluster entropy is very high about 0.69 when $K = 3$ and gets as low as less than 0.1 when K is 5, 7, 9. k-SNAP entropy is always 0 since it groups nodes with the same attribute values into the same clusters.



(a) Density (b) Entropy

Fig. 4. Cluster quality comparison of SAGL and current state-of-the-art methods on DBLP-5000.



(a) Entropy for attribute "prolific" (b) Entropy for attribute "primary topic"

Fig. 5. Cluster quality comparison of SAGL, BAGC, and SA-Cluster on DBLP-10K.

Clustering Results on DBLP-5000. For DBLP-5000, all approaches are executed with community number K=10, 30, 50, 70 for clustering quality comparison. Clustering density and entropy comparison with different values of K are shown in Figure 4. According to Figure 4(a), both SAGL and TSGL have higher density than SA-Cluster in most cases except when $K = 10$. TSGL entropy is around 1.42 or above and higher than SAGL, as shown in Figure 4(b). This occurs because SAGL considers both topological structure and node attributes while TSGL ignores the second factor.

In DBLP-5000, there exist three values for the attribute *prolific* and the 100 values for the attribute *primary topics*. k-SNAP enforces the homogeneity of attributes in each cluster. The topological structures of the detected clusters by k-SNAP on DBLP-5000 are extremely loose, and the densities of these clusters are very low. Compared with other methods, the experimental results of k-SNAP on DBLP-5000 is effectively lower. For this reason, the k-SNAP is not included in Figure 4.

Clustering Results on DBLP-10K. For this dataset, the number of clusters are varied with $K = 50, 100, 200, 300$. We show clustering entropy for each attribute with different K on $\lambda = 0.1$ in Figure 5. According to Figure 5(a), SAGL has lower entropy for

Table IV. Clustering Efficiency on the Political Blogs Dataset

Iter.	K = 3		K = 5		K = 7		K = 9	
	D	E	D	E	D	E	D	E
1	0.90	0.23	0.79	0.14	0.65	0.12	0.54	0.12
2	0.90	0.13	0.72	0.15	0.61	0.15	0.57	0.13
3	0.92	0.15	0.79	0.14	0.63	0.13	0.63	0.14
4			0.89	0.13	0.72	0.14	0.56	0.14
5			0.87	0.14			0.65	0.12

Table V. Clustering Efficiency on the DBLP-5000 Dataset

Iter.	K=10		K=30		K=50		K=70	
	D	E	D	E	D	E	D	E
1	0.57	2.31	0.52	2.14	0.52	1.95	0.52	1.83
2	0.55	1.66	0.53	1.56	0.53	1.50	0.53	1.45
3	0.59	1.37	0.54	1.32	0.54	1.32	0.54	1.30
4	0.59	1.21	0.56	1.22	0.53	1.24	0.53	1.24
5			0.56	1.18	0.54	1.22	0.53	1.23
6					0.54	1.22	0.54	1.21

both attributes. SAGL keeps the entropy around 0.40 for attribute prolific, as shown in Figure 5(a). SAGL obtains about entropy 2.6 for $K = 100, 200, 300$ for attribute "primary topic," as shown in Figure 5(b).

6.5. Clustering Efficiency Evaluation

Cluster efficiency is measured by the convergence speed of the objective function. We show cluster efficiency of our method through gradually increasing the number of iterations with the related values of the cluster density and entropy. Table IV and Table V show the clustering efficiency of SAGL on Political Blogs and DBLP-5000. We show the related cluster density (D) and entropy (E) in each iteration until the objective function converges. In most cases, both density and entropy improve with each iteration. As the value of K increases, more iterations are required for the objective function to converge.

6.6. Time Complexity Analysis

In this section, time complexity analysis is used to evaluate the efficiency of the proposed method SAGL, comparing it with other methods. Our method has three steps: clusters centroids initialization, cluster assignment, and cluster centroid updating, which can be expressed as follows:

$$T_{centroids\_init} + t \cdot \left(T_{cluster\_assign} + T_{centroids\_update}\right).$$

Here t is the number of iterations. The proposed method first calculates PageRank values as a nodes' global importances in the graph. Each node receives a proportion of the PageRank value from its in-neighbors and propagates to its out-neighbors. So the upper-bound complexity is $O(q|V||E|)$, where q is the maximal iteration of PageRank convergence.

To initialize cluster centroids, SAGL selects the top K nodes with time complexity $O(|K|log|K| + |V| - K)$. SAGL costs $O(|E|)$ to calculate edge strength. $T_{centroids\_init}$ costs $O(q|V||E|)$. At each iteration, the time cost of our method to calculate the similarity between each node and the cluster centroids is $O(|N_L|^2)$. If the average number of neighbors of a node is $|E|/|V|$, then the total cluster assignment of all nodes results in a time complexity of $O(|V| \cdot (|E|/|V|)^2)$. Updating the cluster centroids takes $O(|V|^2)$, and the step of attribute weight self-adjustment costs $O(|V|)$.

Table VI. Time Complexity Comparison

Algorithm	Time complexity
SAGL/TSGL	$O(\|V\|\|E\| + \|V\|^2 + \|V\| \cdot (\|E\|/\|V\|)^2)$
SA-Cluster	$O(\|V \cup V_a\|^3)$

The above analysis shows the total time complexity of SAGL to be $O(|V||E| + |V|^2 + |V| \cdot (|E|/|V|)^2)$. If an attributed graph is a complete graph, then the time complexity of SAGL is $O(|V|^3)$. If an attributed graph is not a complete graph, then the time complexity of SAGL is between $O(|V|^2)$ and $O(|V|^3)$. If an attributed graph is sparse, then the time complexity of SAGL is close to $O(|V|^2)$.

Comparing SAGL and TSGL, although TSGL does not compute attribute similarity ($\lambda = 0$), the overall time complexity is the same for both algorithms, because their main steps are the same. The detailed comparison results are shown in Table VI. In Table VI, V_a represents attribute nodes in an attributed augmented graph. SA-Cluster creates an augmented attributed graph to detect communities. For this reason, the node set of the augmented attributed graph is $V \cup V_a$. The largest contribution to the computational complexity of SA-Cluster is a random-walk distance calculation, which requires a series of matrix multiplications and additions. Thus the overall time complexity of SA-Clustering is $O(|V \cup V_a|^3)$.

7. CONCLUSION

In this article, we combined topological structure and node attributes in attributed graphs by analyzing both global structure and local neighbors. We evaluated each node's global importance by utilizing the PageRank algorithm. According to the global importance of a node, we calculated the topological closeness between nodes and then combined it with attribute similarities of nodes. In the clustering process, the similarity between nodes and centroids were evaluated by the similarity of their local neighbors. Our SAGL algorithm exhibited good performance on three real-world datasets.

For future work, we plan to further study the topic model to extract topics from the original description content of a node so we can detect more meaningful topical communities. We attempt to understand how different types of links improve clustering quality.

ACKNOWLEDGMENTS

We thank APIIA International LLC, for their generous research support, and we thank Yang Zhou for providing us with the DBLP dataset.

REFERENCES

Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 US election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*. ACM, 36–43.

Charu C. Aggarwal, Yan Xie, and S. Yu Philip. 2011. Towards community detection in locally heterogeneous networks. In *SDM*. SIAM, 391–402.

Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. 2001. Searching the web. *ACM Trans. Internet Technol.* 1, 1 (2001), 2–43.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

Jie Chen and Yousef Saad. 2012. Dense subgraph extraction with application to community detection. *IEEE Trans. Knowl. Data Eng.* 24, 7 (2012), 1216–1230.

Yi-Cheng Chen, Wen-Yuan Zhu, Wen-Chih Peng, Wang-Chien Lee, and Suh-Yin Lee. 2014. CIM: Community-based influence maximization in social networks. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 25.

Hong Cheng, Yang Zhou, and Jeffrey Xu Yu. 2011. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Trans. Knowl. Discov. Data* 5, 2 (2011), 12.

Juan David Cruz, Cécile Bothorel, and François Poulet. 2013. Community detection and visualization in social networks: Integrating structural and semantic information. *ACM Trans. Intell. Syst. Technol.* 5, 1 (2013), 11.

T. A. Dang and E. Viennet. 2012. Community detection based on structural and attribute similarities. In *Proceedings of the International Conference on Digital Society (ICDS)*. 7–12.

Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2014. Mixing local and global information for community detection in large networks. *J. Comput. Syst. Sci.* 80, 1 (2014), 72–87.

Himel Dev. 2014. A user interaction based community detection algorithm for online social networks. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. ACM, 1607–1608.

Ying Ding, Erjia Yan, Arthur Frazho, and James Caverlee. 2009. PageRank for ranking authors in co-citation networks. *J. Am. Soc. Inform. Sci. Technol.* 60, 11 (2009), 2229–2243.

Michelle Girvan and Mark E. J. Newman. 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.* 99, 12 (2002), 7821–7826.

Leonard Kaufman and Peter Rousseeuw. 1987. *Clustering by Means of Medoids*. North-Holland.

M. E. J. Newman. 2013. Community detection and graph partitioning. *Europhys. Lett.* 103, 2 (2013), 28003.

Mark E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* 74, 3 (2006), 036104.

Athanasios Papadopoulos, George Pallis, and Marios D. Dikaiakos. 2013. Identifying clusters with attribute homogeneity and similar connectivity in information networks. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. IEEE, 343–350.

Sumeet Singh and Amit Awekar. 2013. Incremental shared nearest neighbor density-based clustering. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*. ACM, 1533–1536.

Xuning Tang and Christopher C. Yang. 2014. Detecting social media hidden communities using dynamic stochastic blockmodel with temporal Dirichlet process. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 36.

Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. 2008. Efficient aggregation for graph summarization. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. ACM, 567–580.

Wei Weng, Shunzhi Zhu, and Huarong Xu. 2014. Hierarchical community detection algorithm based on local similarity. *J. Dig. Inform. Manag.* 12, 4 (2014), 275.

Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. 2012. A model-based approach to attributed graph clustering. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 505–516.

Erjia Yan and Ying Ding. 2011. Discovering author impact: A PageRank perspective. *Inform. Process. Manag.* 47, 1 (2011), 125–134.

Jaewon Yang and Jure Leskovec. 2014. Structure and overlaps of ground-truth communities in networks. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 26.

Jaewon Yang, Julian McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM)*. IEEE, 1151–1156.

Xin Yu, Jing Yang, and Zhi-Qiang Xie. 2015. A semantic overlapping community detection algorithm based on field sampling. *Expert Syst. Appl.* 42, 1 (2015), 366–375.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *Eur. Phys. J. B* 71, 4 (2009), 623–630.

Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.* 2, 1 (2009), 718–729.

Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2010. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM)*. IEEE, 689–698.

Received September 2015; revised February 2016; accepted July 2016