entropy

MDPI

Article

Coupled Node Similarity Learning for Community Detection in Attributed Networks

Fanrong Meng [1], Xiaobin Rui [1], Zhixiao Wang [1,*], Yan Xing [2,*] and Longbing Cao [3]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China; mengfr@cumt.edu.cn (F.M.); ruixiaobin@cumt.edu.cn (X.R.)

[2] School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

[3] Advanced Analytical Institute, University of Technology Sydney, Sydney, NSW 2007, Australia; longbing.cao@uts.edu.au

* Correspondence: zhxwang@cumt.edu.cn (Z.W.), xingyan_cumt@163.com(Y.X.)

Received: 22 May 2018; Accepted: 14 June 2018; Published: 17 June 2018

Abstract: Attributed networks consist of not only a network structure but also node attributes. Most existing community detection algorithms only focus on network structures and ignore node attributes, which are also important. Although some algorithms using both node attributes and network structure information have been proposed in recent years, the complex hierarchical coupling relationships within and between attributes, nodes and network structure have not been considered. Such hierarchical couplings are driving factors in community formation. This paper introduces a novel coupled node similarity (CNS) to involve and learn attribute and structure couplings and compute the similarity within and between nodes with categorical attributes in a network. CNS learns and integrates the frequency-based intra-attribute coupled similarity within an attribute, the co-occurrence-based inter-attribute coupled similarity between attributes, and coupled attribute-to-structure similarity based on the homophily property. CNS is then used to generate the weights of edges and transfer a plain graph to a weighted graph. Clustering algorithms detect community structures that are topologically well-connected and semantically coherent on the weighted graphs. Extensive experiments verify the effectiveness of CNS-based community detection algorithms on several data sets by comparing with the state-of-the-art node similarity measures, whether they involve node attribute information and hierarchical interactions, and on various levels of network structure complexity.

Keywords: attributed networks; coupled node similarity; community detection

1. Introduction

Community detection is an important task in complex network analysis. So far, the definition of community is still ambiguous. In most state-of-the-art research, the concept of community is a group of nodes densely connected relatively to the rest of the network. Networks that consider both object interactions and attributes, i.e., *attributed networks*, can be represented by an attributed graph in which nodes represent the objects, edges represent the relationships between objects, and the feature vectors associated with nodes represent the attributes. The network topological structure reflects the interactions between nodes and the node attribute information reflects the common characteristics among nodes. They both play important roles in the formation of the network community structure. However, nowadays most community detection algorithms only use the network topological structure. Community detection on such attributed networks using both network topological structure and node attribute information is important yet challenging, and relies on appropriate similarity learning.

Community detection on attributed networks. Nowadays, many approaches have been proposed that incorporate node attributes and edges in the community detection process.

Existing methods can be classified roughly into two categories. The first category is composed of probabilistic generative models that formulate joint models of edges and node attributes, and that use the models to infer the community memberships of nodes in an attributed network [1–4]. However, they are not as efficient as hybrid methods. Cruz et al. [5] proposed an iterative optimization algorithm by maximizing the modularity and entropy to obtain the structural and semantically related community structures. CODICIL [6] constructs content edges by selecting the top K neighbors of each node using their attributes, obtains the combined similarity of each pair of nodes, and then sparsifies the newly constructed graph with content edges. Finally, an existing community detection algorithm is used to partition the sparsified graph into a given number of communities. SA-cluster [7] views node attributes as virtual vertices, constructs an attribute-augmented graph, and performs a random walk on the attribute-augmented graph to obtain a unified distance. It then adopts the K-medoids algorithm to detect the community based on learned pairwise distance. Inc-cluser [8,9] is a slightly faster version of SA-cluster.

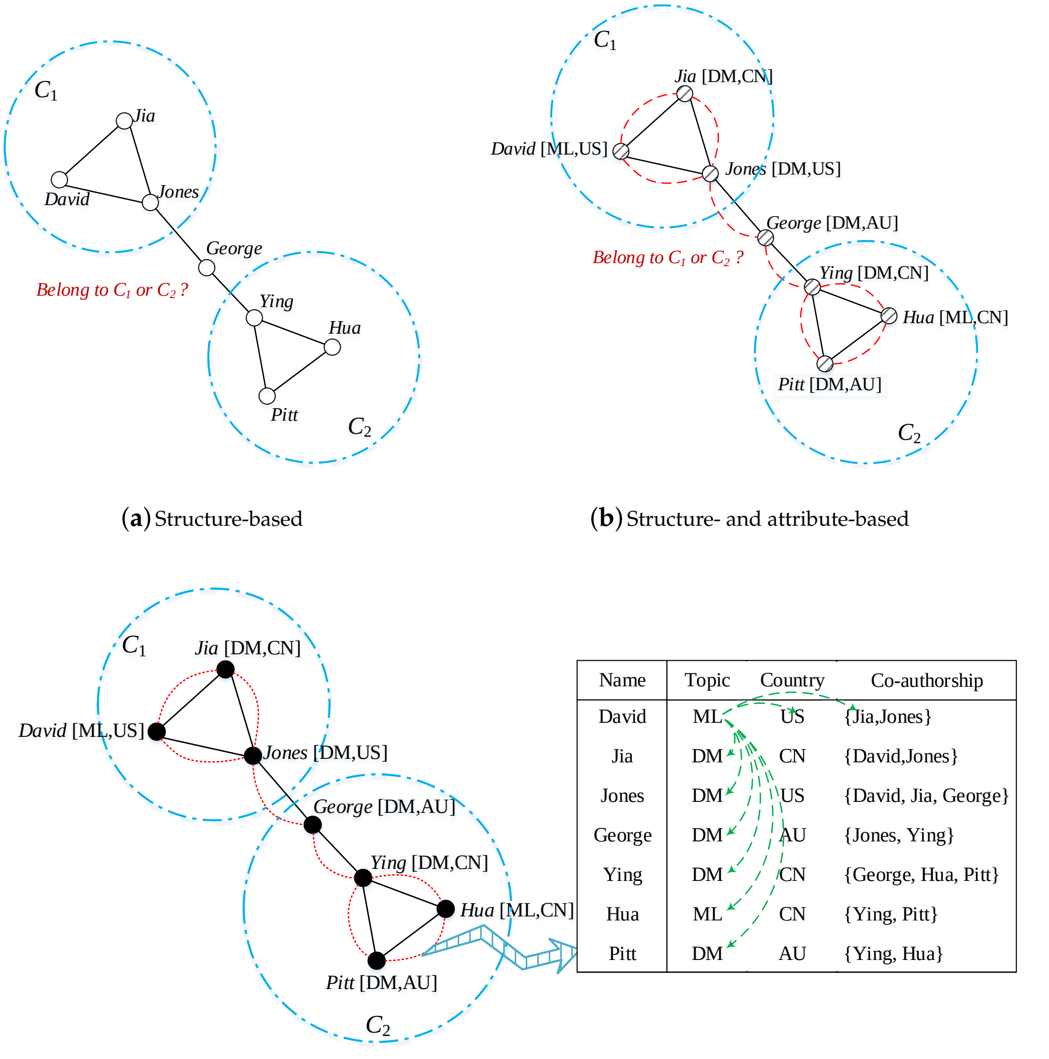
Related work on similarity learning in community detection. In understanding attributed graphs, many methods take the following strategy. First, an attributed graph is converted (reduced) to a weighted graph, where weights represent attribute similarity. The edge weights indicate particularly close connections or similarity between nodes. Then, clustering algorithms for weighted graphs can be applied. For example, Steinhaeuser and Chawla [10] presented a simple approach to constructing a network with edge weights based on node attributes and clustering nodes whose edge weight exceeds the threshold in the same community. The authors show that edge weights based on node attribute similarity are superior to edge weights based on network topology in a large scale-free social network.

Appropriately, learning node similarity is critical for understanding network complexity and effectively detecting communities. Accordingly, a number of methods have been developed that exploit node similarity by capturing node relationships. Node similarity learning in existing methods can be roughly assigned to two categories: *structure similarity* and *attribute similarity*. Structure similarity, which focuses on so-called structure equivalence, is commonly used; that is, two nodes are similar if they share the same or similar network neighbors. Typical examples include the cosine similarity (Cosine) [11] and the Jaccard index (Jaccard) [12]. The other kind of methods collect both local and global scale information to compute the weight of each edge, including k-path edge centrality [13], SimRank-based edge weighting scheme [14], and WNF [15]. Few attribute similarity learning methods are proposed for categorical network data. The representative methods are simple matching coefficient (SMC) [16] and the recently proposed coupled object similarity (COS) [17,18]. COS has been shown to be effective in categorical data analysis for clustering [19], classification [20], and recommender systems [21], as it aims to capture the value-to-object coupling relationships [22] embedded in a complex dataset.

To the best of our knowledge, there are no such node similarity learning methods in attributed network analysis that effectively capture the complex interactions between node attributes and network structure, and the coupling relationships within and between node attributes. These complex interactions and relationships drive the formation of communities, so it is fundamental to understand how they interact and affect community and network dynamics.

Learning complex coupling relationships and our main contributions. We illustrate the various coupling relationships in a co-authoring network in Figure 1, in which the information table shows the co-authoring information. A node represents an author, and an edge represents the co-authoring relationship between two authors. In addition, there is topic and country information associated with each author. We convert the co-authoring information table to graphs based on different approaches to represent the structure and relationships between authors. Below, we discuss the different outcomes of author communities detected as a result of these different representation approaches.

First, if only the structure of the co-authoring graph is considered in the detection of author communities, we have the co-authoring structure shown in Figure 1a. Two clearly separated communities emerge: {David, Jia, Jones} and {Ying, Hua, Pitt}; however, we cannot tell which community George belongs to since he has the same relationship with Jones and Ying who separately belong to two different communities. Second, if both graph structure and node attributes are considered, which results in the diagram in Figure 1b, we still cannot cluster George to a proper community by using SMC to compute the similarity between two connected authors. Since SMC uses 0 and 1 to distinguish the similarity between categorical values, the similarity between authors who live in AU and the US is equal to that between authors who live in AU and CN. Therefore, the similarity between George and Jones is still same as the similarity between George and Ying. However, by involving the co-authoring relationships in Figure 1c, we observe that the similarity between AU and CN should be greater because authors from these two countries collaborate more frequently. Therefore, George is more similar to Ying than Jones, and we can correctly divide George to the right community.



(**a**) Structure-based

(**b**) Structure- and attribute-based

Name	Topic	Country	Co-authorship
David	ML	US	{Jia,Jones}
Jia	DM	CN	{David,Jones}
Jones	DM	US	{David, Jia, George}
George	DM	AU	{Jones, Ying}
Ying	DM	CN	{George, Hua, Pitt}
Hua	ML	CN	{Ying, Pitt}
Pitt	DM	AU	{Ying, Hua}

(**c**) Coupled structure- and attribute-based

Figure 1. Structure and attribute couplings in a co-authoring network. (Note: Symbol — indicates the linkage between two nodes; - - - - refers to simple attribute similarity between two nodes, and ⋯⋯⋯ represents the complex coupling relationships between two nodes.)

The above three scenarios illustrate the importance of involving relevant information and relationships and learning their similarity in community detection. As shown in the limited work reported in the literature [23], engaging both structure and attribute similarities can generate more meaningful communities. However, existing methods do not consider the complex interactions within and between attributes, and between node attributes and structure. In most attributed networks, nodes prefer to connect to other nodes with similar attributes (i.e., homophily) [24]. The presence of homophily has been discovered in a vast array of network studies. More than 100 studies that have observed homophily in some form or another and they establish that similarity breeds connection [25]. The homophily property reflects the effect of node attributes on the network edges. On the other hand, the edges in the network should also reflect the difference between attributes. In this paper, we propose a novel coupled node similarity (CNS) learning method, which involves both node attributes and structure information in an attributed graph. The main idea behind CNS and its contributions to community detection are presented below:

- CNS captures different levels of coupling relationships in an attributed graph, including value-to-value, value-to-node, and attribute-to-structure relationships. To the best of our knowledge, this is the first work that systematically represents the hierarchical interactions in terms of both structural and attribute aspects.
- CNS learns the above respective relationships in terms of calculating and integrating the intra-attribute coupled similarity, the inter-attribute coupled similarity, and the coupled attribute-to-structure similarity. Hence, CNS captures not only the attribute value interactions within and between attributes, but also the interactions between node attributes and structure. This provides a comprehensive means of understanding the intrinsic driving forces and complexity in community formation.
- We incorporate CNS into attributed graphs to generate weighted graphs, combining the topological structure and node attributes in a unified manner to detect communities in attributed networks.
- We also empirically evaluate the effectiveness of CNS similarity in terms of whether node attributes are involved, what types of node interactions are learned, and different levels of network structure complexity.

2. Learning Coupled Node Similarity

In this section, we introduce the framework and specific similarity measures for learning coupled node similarity.

2.1. The CNS Framework

The framework for learning CNS is shown in Figure 2. CNS captures four sources of interactions and similarities: (1) the *intra-attribute coupled similarity* learns the interactions within a node attribute; (2) the *inter-attribute coupled similarity* models the interactions between node attributes; (3) the *coupled attribute similarity* integrates both of them; and (4) the *coupled attribute-to-structure similarity* captures the interactions between node attributes and network structure. Lastly, CNS integrates the coupled attribute similarity and the coupled attribute-to-structure similarity to represent the overall relationships and similarities in an attributed network.

An attributed network can be modeled as a graph $G = (V, E, F)$, where V is the set of nodes, E is the set of edges, and F is the set of node attribute vectors. All the main notations are described in Table 1.

Table 1. Notation explanation.

Notation	Description
M	The number of node attributes
(F_m)	The set of all distinct values on the mth attribute
$F_m(i)$	The value of the mth attribute for node i
K	The number of communities
C	The communities of the network, $C = \bigcup_{k=1}^{K} C_k$
c_i	The community to which node i belongs
Γ_i	The neighbor set of node i
$A(i,j)$	The adjacency relationship between nodes i and j. $A(i,j) = 1$ if nodes i and j are connected; otherwise $A(i,j) = 0$
$W(i,j)$	The weight between nodes i and j
$S(i,j)$	The similarity between nodes i and j
$l_r(i)$	The received label of node i
m^w	The sum of all edge weights in the network, $m^w = \sum_{i,j \in V} W(i,j)$
d_i^w	The sum of edge weights which are connected to node i, $d_i^w = \sum_{j \in \Gamma_i} W(i,j)$
$g_m(x)$	The node set whose mth attribute value is x
α_n	The weight parameter for the nth attribute, $\sum_{n=1}^{M} \alpha_n = 1$, $\alpha_n \in [0,1]$
$\delta_{m\|n}(x,y)$	the inter-relative attribute coupled similarities between values x and y of the mth attribute based on the nth attribute ($n \neq m$)
$\overline{B}$	$\overline{B} = F_n / B$, the complement set of B under the complete distinct value set F_n of the nth attribute
$g_n^*(B)$	the node set whose attribute value in the nth attribute is in B
$\delta_m^{Ia}(x,y)$	The intra-attribute coupled similarity between the attribute values x and y of the mth attribute
$\delta_m^{Ie}(x,y)$	The inter-attribute coupled similarity between the attribute values x and y of the mth attribute based on other attributes
$\delta_m^{A}(x,y)$	The coupled attribute similarity between the attribute values x and y of the mth attribute
$\delta_m^{AS}(x,y)$	The coupled attribute-to-structure similarity between the attribute values x and y of the mth attribute
$CAS(i,j)$	The coupled attribute similarity between nodes i and j
$CNS(i,j)$	The coupled node similarity between nodes i and j

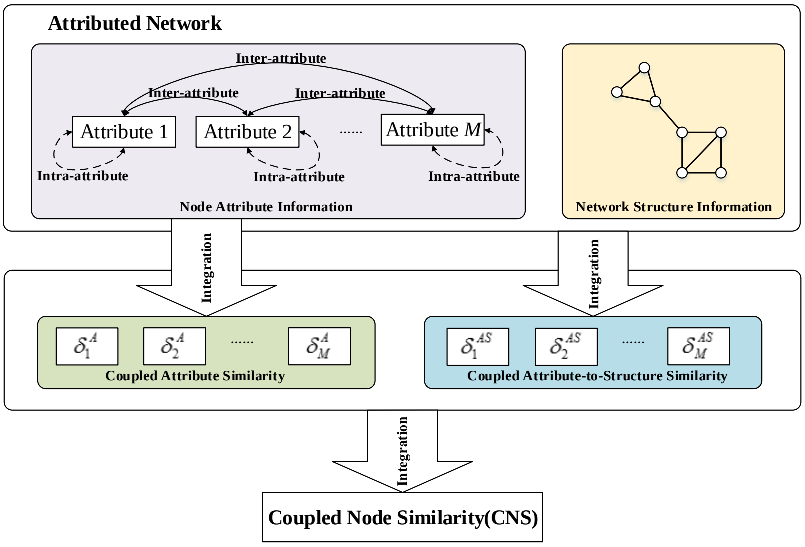


Figure 2. The framework for learning CNS. (Note: Symbol ←----→ indicates intra-attribute coupled similarity calculated using the interaction between attribute values within an attribute and $\longleftrightarrow$ refers to inter-attribute coupled similarity involved the couplings between attributes. The coupled attribute similarity in the second level integrates both of intra-attribute coupled similarity and inter-attribute coupled similarity. The coupled attribute-to-structure similarity in the second level captures the interactions between node attributes and network structure. In the last level, CNS integrates the coupled attribute similarity and the coupled attribute-to-structure similarity.)

2.2. Coupled Attribute Similarity

Coupled attribute similarity (CAS) is extended from the concept of Coupled Attribute Similarity for Object (CASO) in Wang et al. [18]. CASO is based on the coupled attribute similarity for values, by considering both the intra-coupled and inter-coupled attribute value similarities, which globally capture the attribute value frequency distribution and attribute dependency aggregation with high accuracy and relatively low complexity. CAS combines the intra-attribute coupled similarity (Defintion 1) and inter-attribute coupled similarity (Defintion 2) to cater for specific characteristics in network data.

Definition 1. *(Intra-Attribute Coupled Similarity) The intra-attribute coupled similarity $\delta_m^{Ia}(x, y)$ between node attribute values where x and y, x= $F_m(i)$ and y = $F_m(j)$ are the values of nodes i and j in the mth attribute, is calculated by considering the relationship between the frequency of their occurrence.*

$$\delta_m^{Ia}(x, y) = \frac{|g_m(x)| \times |g_m(y)|}{|g_m(x)| + |g_m(y)| + |g_m(x)| \times |g_m(y)|} \tag{1}$$

$g_m(x)$ and $g_m(y)$ are the node sets which have the same attribute value as nodes i and j, respectively, in the mth attribute. $|g_m(x)|$ and $|g_m(y)|$ are the occurrence times of node attribute values x and y across all nodes in the network.

In the toy example in Figure 1, for example, there are two authors from Australia {George, Pitt} and three from China {Ying, Hua, Jia}, so $\delta_{country}^{Ia}(AU, CN) = 6/11$.

Below, the *inter-attribute coupled similarity* is defined, which considers the couplings between node attributes when the node attribute value similarity is calculated.

Definition 2. *(Inter-Attribute Coupled Similarity) The inter-attribute coupled similarity $\delta_m^{Ie}(x, y)$ between values x and y of the mth attribute based on other attributes is defined as follows.*

$$\delta_m^{Ie}(x, y) = \sum_{n=1, n \neq m}^{M} \alpha_n \delta_{m|n}(x, y). \tag{2}$$

α_n is the weight parameter for the nth attribute, $\sum_{n=1}^{M} \alpha_n = 1$, $\alpha_n \in [0, 1]$. M is the total number of node attributes. $\delta_{m|n}(x, y)$ is one of the inter-relative attribute coupled similarities between values x and y of the mth attribute based on the nth attribute ($n \neq m$).

$$\delta_{m|n}(x, y) = \min_{B \subseteq F_n} \{2 - P_{n|m}(B|x) - P_{n|m}(\overline{B}|y)\}. \tag{3}$$

F_n represents the attribute values on the nth attribute. B is a subset of attribute values on the nth attribute. $\overline{B} = F_n / B$ is the complement set of B under the complete distinct value set F_n of the nth attribute. $P_{n|m}(B|x)$ is the information conditional probability (ICP) of B with respect to x, which is defined as follows.

$$P_{n|m}(B|x) = \frac{|g_n^*(B) \cap g_m(x)|}{|g_m(x)|}. \tag{4}$$

$g_n^*(B)$ is the node set whose attribute value in the nth attribute is in B. Intuitively, when given all the objects with the value x on mth attribute, ICP is the percentage of common objects whose values on the nth attribute fall in subset B and whose values on the mth attribute are exactly x as well.

In the toy example in Figure 1, $F_{topic} = \{DM, ML\}$, and the number of its power sets is four. If $B = \{DM\}$ then $\overline{B} = \{ML\}$, $g_{topic}^*(B) = \{Jia, Jones, George, Ying, Pitt\}$, $g_{country}(AU) = \{George, Pitt\}$, $P_{topic|country}(B|AU) = 1$. Similarly, $P_{topic|country}(\overline{B}|CN) = 1/3$. Considering all conditions, $\delta_{country|topic}(AU, CN) = 2/3$. Since there are only two attributes, $\delta_{country}^{Ie}(AU, CN) = 2/3$.

Definition 3. *(Coupled Attribute Similarity) The coupled attribute similarity $\delta_m^A(x,y)$ between values x and y of the mth attribute is the combination of the intra-attribute coupled similarity and the inter-attribute coupled similarity between x and y.*

$$\delta_m^A(x,y) = \delta_m^{Ia}(x,y) \times \delta_m^{Ie}(x,y). \tag{5}$$

Lastly, the *coupled attribute similarity (CAS)* for the two nodes i and j is calculated as follows.

$$CAS(i,j) = \sum_{m=1}^{M} \delta_m^A(F_m(i), F_m(j)). \tag{6}$$

2.3. Coupled Attribute-to-Structure Similarity

In an attributed network, not all node attributes are equally important for community detection; even for an attribute, two different value pairs may not contribute the same. Based on the homophily property [24] of social networks, i.e., nodes to be connected with other nodes that share similar attributes, the consistency between node attributes and structure information could guide the community detection process. Therefore, the coupled attribute-to-structure similarity is proposed to measure the different contribution of different attribute value pairs.

Definition 4. *(Coupled Attribute-to-Structure Similarity) The coupled attribute-to-structure similarity $\delta_m^{AS}(x,y)$ between values x and y of the mth attribute is defined as the degree of consistency between the attribute value pair (x,y) and the linkage across all nodes in the network. It is equal to the number of edges between the two node sets whose attribute values are x and y, respectively, in the mth attribute divided by the total number of possible edges between them.*

$$\delta_m^{AS}(x,y) = \frac{\sum_{v_1 \in g_m(x), v_2 \in g_m(y)} A(v_1, v_2)}{|g_m(x)| \times |g_m(y)|}. \tag{7}$$

In the toy example in Figure 1, there are two authors from Australia and three from China, and there are three connections between the authors from these two countries, {George-Ying, Pitt-Ying, Pitt-Hua}, so $\delta_{country}^{AS}(AU, CN) = 0.5$.

2.4. Coupled Node Similarity

Coupled node similarity is defined as the combination of the coupled attribute similarity and coupled attribute-to-structure similarity.

Definition 5. *(Coupled Node Similarity) The coupled node similarity $CNS(i,j)$ between nodes i and j is calculated below:*

$$\begin{aligned} CNS(i,j) &= \sum_{m=1}^{M} \delta_m^A(F_m(i), F_m(j)) \times \delta_m^{AS}(F_m(i), F_m(j)) \\ &= \sum_{m=1}^{M} \delta_m^{Ia}(F_m(i), F_m(j)) \times \delta_m^{Ie}(F_m(i), F_m(j)) \times \delta_m^{AS}(F_m(i), F_m(j)) \end{aligned}. \tag{8}$$

In the toy example in Figure 1, $CNS(George, Ying) = 0.41$, $CNS(George, Jones) = 0.29$, and George is more similar to Ying, so he belongs to community C_2.

2.5. The Algorithm for Learning CNS

Algorithm 1 presents the process of learning coupled node similarity (CNS). It first calculates the coupled attribute similarity and the coupled attribute-to-structure similarity for all attribute value pairs (Lines 1–8) and then computes CNS for all nodes (Lines 9–17). The CASS function computes the coupled attribute-to-structure similarity (Lines 19–24).

Algorithm 1 Learning Coupled Node Similarity

```
Input: $G(V, E, F)$
Output: $CNS$
 1: for $m \leftarrow 1, M$ do
 2:     for all value pairs $x, y \in$ unique$(F_m)$ do
 3:         $\delta_m^{Ia}(x, y) = CIAAS(x, y, m)$
 4:         $\delta_m^{Ie}(x, y) = CIEAS(x, y, m)$
 5:         $\delta_m^{A}(x, y) = \delta_m^{Ia}(x, y) \times \delta_m^{Ie}(x, y)$
 6:         $\delta_m^{AS}(x, y) = CASS(x, y, m)$
 7:     end for
 8: end for
 9: for all nodes $i$ and $j \in V$ do
10:     for $m \leftarrow 1, M$ do
11:         $x = F_m(i)$ , $y = F_m(j)$
12:         if $A(i, j) == 1$ then
13:             $CNS(i, j) += \delta_m^{A}(x, y) \times \delta_m^{AS}(x, y)$
14:         end if
15:     end for
16: end for
17: return $CNS$
18:
19: Function $CASS(x, y, m)$
20: for all nodes $v_1 \in g_m(x)$ and $v_2 \in g_m(y)$ do
21:     $Enum += A(v_1, v_2)$
22: end for
23: $\delta_m^{AS}(x, y) = Enum/(|g_m(x)| \times |g_m(y)|)$
24: return $\delta_m^{AS}(x, y)$
25:
26: Function $CIAAS(x, y, m)$
27: $U_1 \leftarrow \{v_i | F_m(i) == x\}, U_2 \leftarrow \{v_i | F_m(i) == y\}$
28: $\delta_m^{Ia}(x, y) = (|U_1| \times |U_2|)/(|U_1| + |U_2| + |U_1| \times |U_2|)$
29: return $\delta_m^{Ia}(x, y)$
30:
31: Function $CIEAS(x, y, m)$
32: $U_1 \leftarrow \{v_i | F_m(i) == x\}, U_2 \leftarrow \{v_i | F_m(i) == y\}$
33: for $(n \leftarrow 1, M) and (n \neq m)$ do
34:     for all subset $B \in F_m$ do
35:         $U_3 \leftarrow \{v_i | F_m(i) \in B\}, U_4 \leftarrow \{v_i | v_i | F_m(i) \in (F_m - B)\}$
36:         $ICP_x(B) = (|U_1| \bigcap |U_3|)/(|U_1|)$
37:         $ICP_y(F_m - B) = (|U_2| \bigcap |U_4|)/(|U_2|)$
38:     end for
39:     $Min_{m|n} = min(2 - ICP_x - ICP_y)$
40:     $\delta_m^{Ie}(x, y) += \alpha_n \times Min_{m|n}$
41: end for
42: return $\delta_m^{Ie}(x, y)$
```

2.6. Complexity Analysis

CNS integrates three similarities, e.g., The intra-attribute coupled similarity, the inter-attribute similarity, and the coupled attribute-to-structure similarity. The time complexity analysis is as follows: (1) Compute intra-attribute coupled similarity: $O(MR^2|V|)$, where $|V|$ is the number of nodes in the network; (2) Compute inter-attribute coupled similarity: $O(M^2R^22^R|V|)$, where R is the maximal number of values for each attribute and M is the number of node attributes; (3) Compute coupled attribute-to-structure similarity: $O(MR^2|V|)$. Therefore, the overall time complexity is $O(M^2R^22^R|V|)$.

3. Similarity-Based Community Detection

Our proposed method mainly concentrates on unweighted graphs. CNS is used to generate the edge weight ($W(i,j)$) where an edge exists when two nodes are linked structurally.

$$W(i,j) = \begin{cases} S(i,j) & if\ A(i,j) = 1 \\ 0 & otherwise \end{cases}. \tag{9}$$

$S(i,j)$ represents a similarity metric (e.g., $CNS(i,j)$) to be used to construct the weighted network. SLPA [26], BGLL [27], and K-medoids [28] then detect communities on the weighted networks.

SLPA is an extension of LPA [29] that can analyze communities in weighted networks. It starts by giving each node a unique label and provides each node with a memory to store received labels. In every iteration, each node receives labels from its neighbors and adds the most popular label to its memory. The most popular label is that which carries the maximum weight according to nodes that send the same label. Lastly, every node chooses the maximum frequent label in its memory as its community label and nodes with the same label are assigned to one community.

$$l_r(i) = \underset{l}{\operatorname{argmax}} \sum_{j\in\Gamma_i} W(i,j) \times \varphi(l_s(j), l). \tag{10}$$

$l_r(i)$ represents the received label of node i and $l_s(j)$ is the send label from node j. If $l_s(j) = l$, then $\varphi(l_s(j), l) = 1$, else $\varphi(l_s(j), l) = 0$.

BGLL is an iterative two-phase algorithm based on weighted modularity (WQ) optimization. In the first phase, all nodes are placed into different communities. For each node i, BGLL considers each neighbor j and evaluates the gain of WQ that would take place if i was removed from its community and placed in the community of j. Node i is then placed in the community for which this gain is maximum and positive. The second phase consists of building a new network whose nodes are now the communities found during the previous phase, and the weights of the edges between the new nodes are given by the sum of the weight of the edges between nodes in the corresponding two communities.

$$WQ = \frac{1}{m^w} \sum_{i,j\in V} [W(i,j) - \frac{d_i^w d_j^w}{m^w}] \times \varphi(c_i, c_j) \tag{11}$$

$m^w = \sum_{i,j\in V} W(i,j)$, $d_i^w = \sum_{j\in\Gamma_i} W(i,j)$, c_i and c_j respectively denote the community to which nodes i and j belong. If $c_i = c_j$, then $\varphi(c_i, c_j) = 1$, else $\varphi(c_i, c_j) = 0$.

K-medoids is a clustering algorithm related to the K-means algorithm [30]. Its inputs are the similarity matrix and the number of clusters K. In our experiments, K is set to the true number of clusters. The similarity between two connected nodes is equal to the edge weight that connects them, and the similarity of two disconnected nodes is 0. First, it selects K initial medoids randomly; clusters are then defined as the subsets of points that are similar to the respective medoids, and the objective function is defined as the similarity between a point and the corresponding medoid. The new medoids are then updated as the object of a cluster whose average similarity to all the objects in the cluster is maximal. This process is repeated until all medoids no longer change.

4. Experiments and Analysis

Similarity measures for comparison. This section compares CNS with several representative node similarity measures including Adjacency, Cosine, Jaccard, SMC, and CAS in terms of community detection performance. Table 2 shows the main formulas.

Table 2. The similarities.

Similarity	Formula
Adjacency	$S_{Adjacency}(i,j) = A(i,j)$
Cosine	$S_{Cosine}(i,j) = \frac{\lvert\Gamma_i \cap \Gamma_j\rvert}{\sqrt{\lvert\Gamma_i\rvert \times \lvert\Gamma_j\rvert}}$
Jaccard	$S_{Jaccard}(i,j) = \frac{\lvert\Gamma_i \cap \Gamma_j\rvert}{\lvert\Gamma_i \cup \Gamma_j\rvert}$
SMC	$S_{SMC}(i,j) = \frac{\sum_{m=1}^{M} 1(F_m(i)=F_m(j))}{M}$
CAS	Equation (6)
CNS	Equation (8)

Baseline methods. SLPA, BGLL, and *K*-medoids are used. Since SLPA and *K*-medoids are not stable, they are repeated 100 times and averaged for the final results. The value of parameter α_n in CAS and CNS is $1/M$. M is the total number of node attributes. We apply the algorithms on both synthetic and real networks to test their community detection performance.

Synthetic networks. The structure-only networks consisting of nodes, edges, and communities are generated according to the LFR benchmark networks [31], which are currently the most commonly used synthetic networks in community detection. An LFR network includes the following parameters: N is the number of nodes; $avgk$ is the average degree of the nodes; $maxk$ is the maximum degree of the nodes; $minc$ is the number of nodes contained by the minimum community; $maxc$ is the number of nodes contained by the biggest community; mu is a mixed parameter, which is the probability of nodes connected to nodes of an external community. The greater mu is, the more difficult it is to detect the community structure.

In real networks, not all node attributes are the same important for community detection. Some are critical for cluster nodes, and some are not as important or are not even relevant. Therefore, three kinds of value distributions are generated as follows. (1) Attribute 1: For each community, all of the nodes in a community are assigned the same domain value; (2) Attribute 2: All of the nodes in the network are assigned a random domain value; (3) Attribute 3: All of the nodes in each community are assigned the same domain value. Nodes in the community are selected to host the noise. The noise is a random domain value that is different from the cluster domain value. The noise level nl (the percentage of noise nodes) can be varied.

Real networks. Experiments are also conducted on three well-known real networks: the lawyer friendship network (Lazega) [32], the researcher relationship network (Research) [33], and the counselor relationship network (Consult) [33]. The detailed information of each network is shown in Table 3.

Table 3. The information of real networks.

ID	Name	Abbr.	$\lvert V\rvert$	$\lvert E\rvert$	K	M
R1	Lazega	Laz	71	575	2	7
R2	Research	Res	77	2228	3	4
R3	Consult	Con	46	879	4	2

$\lvert V\rvert$:The number of nodes; $\lvert E\rvert$: The number of edges; K: The number of communities; M: The number of node attributes.

Lazega reflects corporate law partnership in a Northeastern US corporate law firm from 1988 to 1991. It includes friendship and working networks between the 71 attorneys of this firm. Various number of attributes are used in this paper, including status (1: partner; 2: associate), gender (1: man; 2: woman), office (1: Boston; 2: Hartford; 3: Providence), years with the firm, age, practice (1: litigation; 2: corporate), and law school (1: harvard, yale; 2: ucon; 3: other).

Research is about a research team consisting of 77 employees in a manufacturing company. The dataset contains several attributes of each employee: location (1: Paris; 2: Frankfurt; 3: Warsaw; 4: Geneva), tenure (1: 1–12 months; 2: 13–36 months; 3: 37–60 months; 4: 61+ months), and the organizational level (1: Global Dept Manager; 2: Local Dept Manager; 3: Project Leader; 4: Researcher). Since the network is a weighted and directed network, we first convert it to an unweighted and undirected network.

Consult is the relationship between 46 employees in a consulting company. The following attributes are known for the counselors: the organisational level (1: Research Assistant; 2: Junior Consultant; 3: Senior Consultant; 4: Managing Consultant; 5: Partner), gender (1: male; 2: female), region (1: Europe; 2: USA), and location (1: Boston; 2: London; 3: Paris; 4: Rome; 5: Madrid; 6: Oslo; 7: Copenhagen). This network is also a weighted and directed network, we first convert it to an unweighted and undirected network.

Evaluation Criteria. For networks with known community structure, we use normalized mutual information (NMI) [34], F-Measure [35] and Accuracy as the evaluation criteria to compare results of different algorithms. The calculation formulas are shown as follows.

$$NMI = \frac{-2 \times \sum_{r=1}^{R} \sum_{k=1}^{K} \frac{|U_r \bigcap C_k|}{|V|} \log(\frac{|V| \times |U_r \bigcap C_k|}{|U_r| \times |C_k|})}{\sum_{r=1}^{R} \frac{|U_r|}{|V|} \log(\frac{|U_r|}{|V|}) + \sum_{k=1}^{K} \frac{|C_k|}{|V|} \log(\frac{|C_k|}{|V|})}. \tag{12}$$

$C = \{C_1, C_2, \cdots, C_K\}$ represents a community detection result generated by the evaluated algorithm, and $U = \{U_1, U_2, \cdots, U_R\}$ represents the ground-truth community structure. $|V|$ represents the number of nodes in the network. K and R are the number of communities.

$$F - Measure = \sum_{r=1}^{R} \frac{|U_r|}{|V|} \max_{C_k \in C} F(U_r, C_k). \tag{13}$$

$$F(U_r, C_k) = \frac{2 \times P(U_r, C_k) \times R(U_r, C_k)}{P(U_r, C_k) + R(U_r, C_k)}. \tag{14}$$

$P(U_r, C_k) = |U_r \bigcap C_k| / |C_k|$, and $R(U_r, C_k) = |U_r \bigcap C_k| / |U_r|$.

$$Accuracy = TC/|V|. \tag{15}$$

TC represents the number of correct clustering nodes.

4.1. Detection Performance with vs. without Node Attribute Information

This section performs experiments to compare the results of three algorithms based on different similarity methods that do or do not involve node attribute information. The results are shown in Tables 4–6. Numbers in bold style means they are the biggest among six similarities.

Tables 4–6 show that community detection based on CNS achieves better NMI (e.g., maximally 35.45% improvement on the Consult data), F-Measure (e.g., maximally 12.14% improvement on the Consult data), and accuracy (e.g., maximally 15.14% improvement on the Consult data) when compared with the best result of other structure and attribute similarity measures. The results based on SMC are not always better than those based on structure similarities. This illustrates the importance of considering the complex hierarchical interactions within and between node attributes and network structure when calculating node similarity. When the similarity based solely on the node attribute is compared, CAS cannot guarantee better results than SMC. This means the interactions between node attributes and network structure play a vital role in capturing node similarity.

Table 4. The results of NMI(%) w.r.t. six similarities.

Similarity	**SLPA**			**BGLL**			***K*-Medoids**		
	Laz	**Res**	**Con**	**Laz**	**Res**	**Con**	**Laz**	**Res**	**Con**
Adjacency	14.04	65.68	58.15	31.47	70.92	**66.42**	11.25	29.02	20.15
Cosine	25.81	80.22	64.94	36.65	75.66	49.60	22.61	58.64	37.79
Jaccard	26.16	78.48	64.66	39.13	75.62	49.60	39.76	62.88	30.80
SMC	27.67	92.05	70.46	39.04	**100**	**66.42**	28.46	34.75	47.74
CAS	26.11	87.84	67.23	36.18	86.64	**66.42**	72.55	34.08	54.11
CNS	**29.47**	**98.71**	**78.67**	**48.67**	**100**	**66.42**	**76.02**	**76.46**	**73.29**
Δ%	6.51	7.24	11.65	24.38	0.00	0.00	4.78	21.60	35.45

Table 5. The results of F-Measure(%) w.r.t. six similarities.

Similarity	**SLPA**			**BGLL**			***K*-Medoids**		
	Laz	**Res**	**Con**	**Laz**	**Res**	**Con**	**Laz**	**Res**	**Con**
Adjacency	52.32	68.25	80.00	71.96	62.98	**80.87**	53.54	40.36	63.36
Cosine	66.27	87.22	79.81	75.85	89.52	36.67	71.57	70.58	76.66
Jaccard	65.32	85.00	79.62	70.14	86.47	36.67	75.21	74.85	74.52
SMC	65.57	93.89	74.11	73.81	**100**	**80.87**	74.48	54.06	74.48
CAS	64.84	92.60	82.12	77.39	71.55	**80.87**	91.60	52.40	75.78
CNS	**67.38**	**98.93**	**90.79**	**80.41**	**100**	**80.87**	**93.48**	**78.80**	**85.97**
Δ%	1.67	5.37	10.56	3.90	0.00	0.00	2.05	5.28	12.14

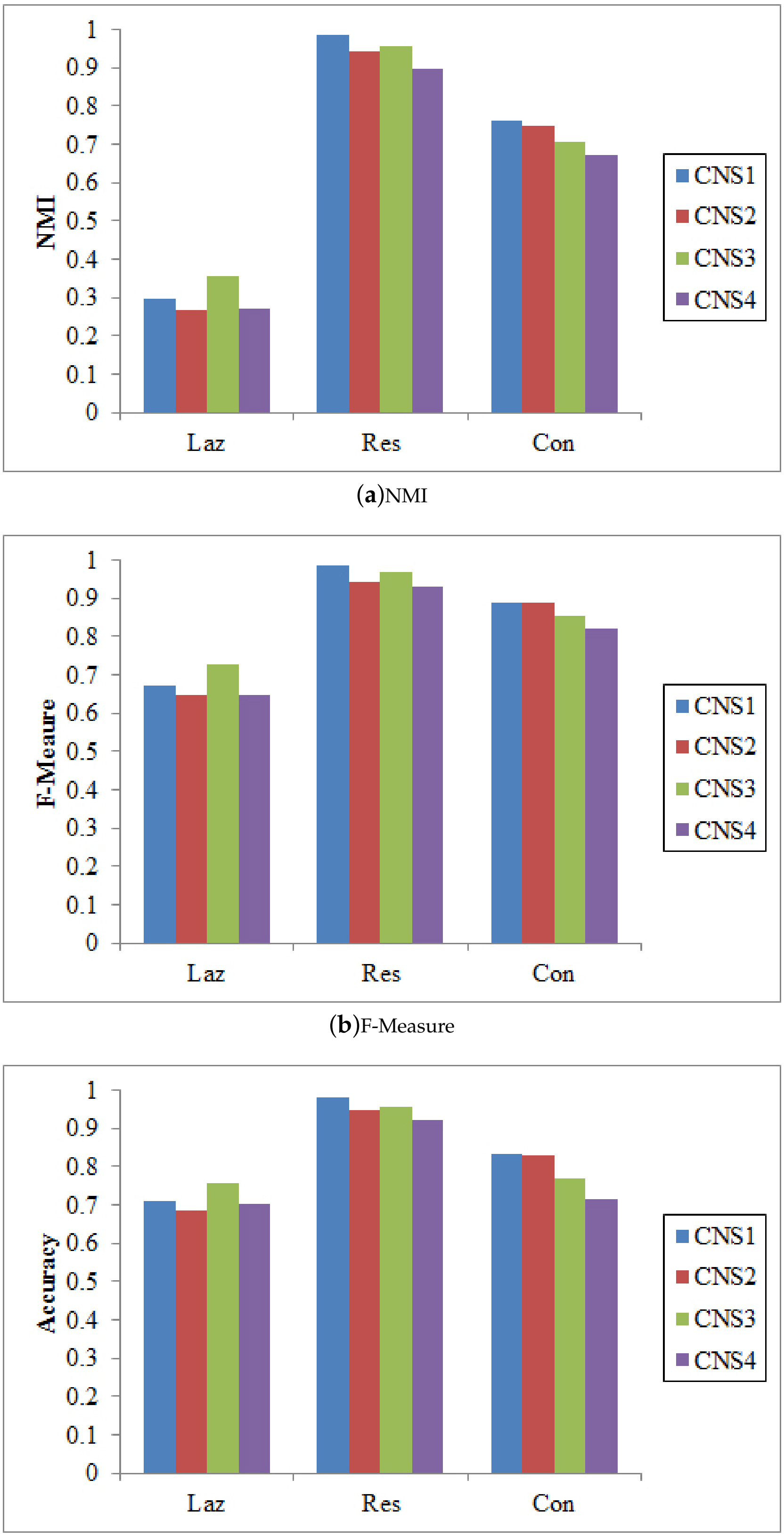
Table 6. The results of Accuracy(%) w.r.t. six similarities.

Similarity	**SLPA**			**BGLL**			***K*-Medoids**		
	Laz	**Res**	**Con**	**Laz**	**Res**	**Con**	**Laz**	**Res**	**Con**
Adjacency	61.32	73.95	72.60	60.87	66.22	**70.00**	60.14	46.31	67.10
Cosine	63.16	93.47	68.35	63.77	86.49	27.50	72.62	71.55	77.05
Jaccard	59.64	80.35	68.10	55.07	78.38	27.50	76.14	75.97	75.03
SMC	68.22	89.89	74.95	59.42	**100**	**70.00**	75.20	56.92	76.60
CAS	70.26	91.65	71.78	66.67	83.78	**70.00**	92.17	54.03	78.03
CNS	**70.87**	**98.36**	**86.30**	**69.57**	**100**	**70.00**	**93.64**	**80.96**	**88.10**
Δ%	0.87	5.23	15.14	4.35	0.00	0.00	1.59	6.57	12.91

4.2. Effect of Differently Integrating Node Similarities

There are different ways to integrate the proposed node similarity components to form the coupled node similarity. Four combinations are used to obtain CNS: $CNS1 = \delta_m^{Ia} \times \delta_m^{Ie} \times \delta_m^{AS}$, $CNS2 = (\delta_m^{Ia} + \delta_m^{Ie}) \times \delta_m^{AS}$, $CNS3 = \delta_m^{Ia} \times \delta_m^{Ie} + \delta_m^{AS}$, and $CNS4 = \delta_m^{Ia} + \delta_m^{Ie} + \delta_m^{AS}$. These CNSs are then fed into SLPA, BGLL, and *K*-medoids for community detection.

Figures 3–5 show that CNS1 and CNS3 are better in most cases, e.g., CNS1 gains 42.40% improvement of NMI over CNS4 on the Lazega data, and CNS3 gains 47.41% improvement over CNS4. However, we cannot tell which works the best in all cases. Various combinations of the three types of similarities may lead to different results and sometimes the difference is significant (e.g., NMI between 54.21% and 76.02% on the Lazega data). This will be further explored in our future work.

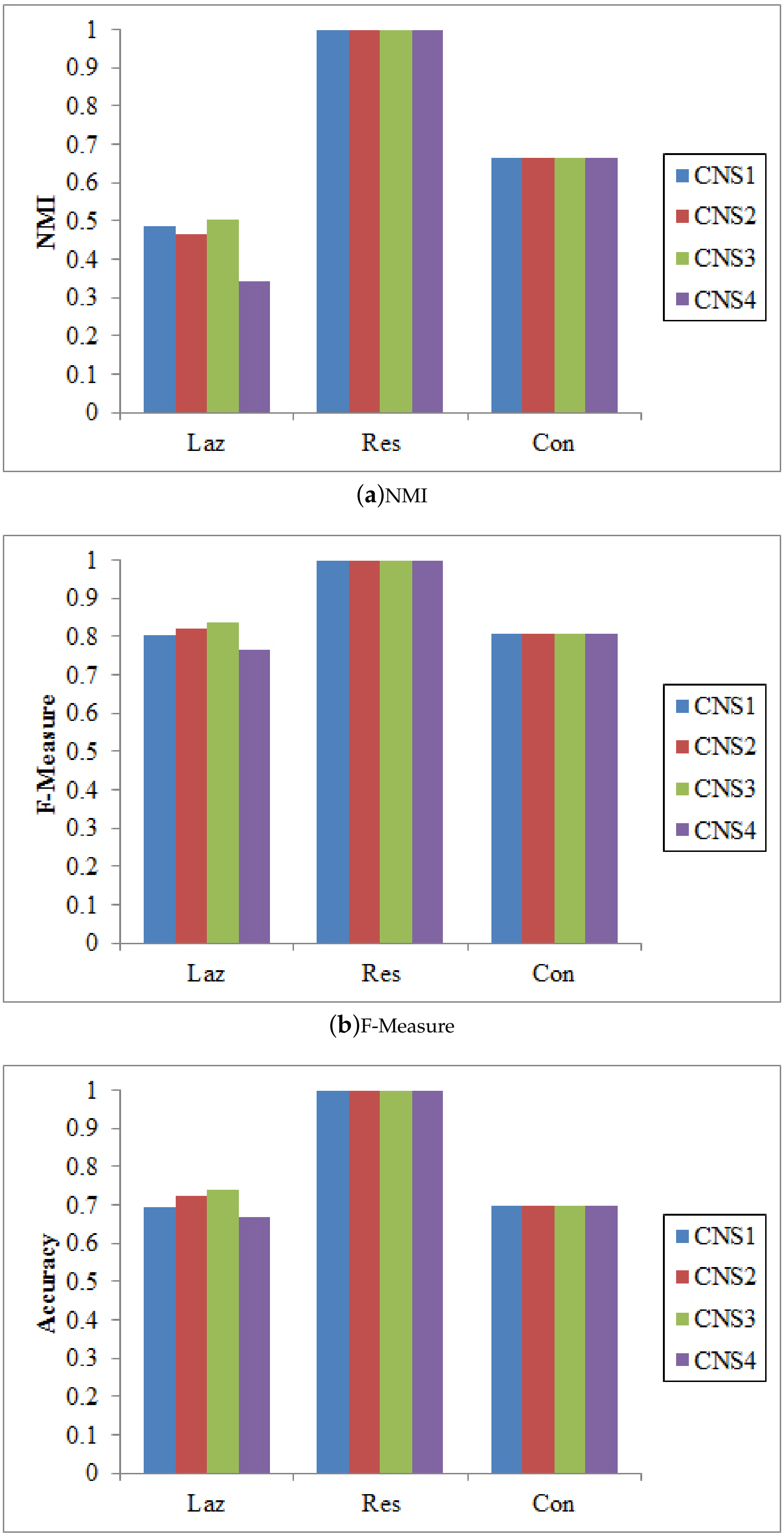


(**a**)NMI

(**b**)F-Measure

(**c**)Accuracy

Figure 3. The results of SLPA w.r.t. different CNSs.

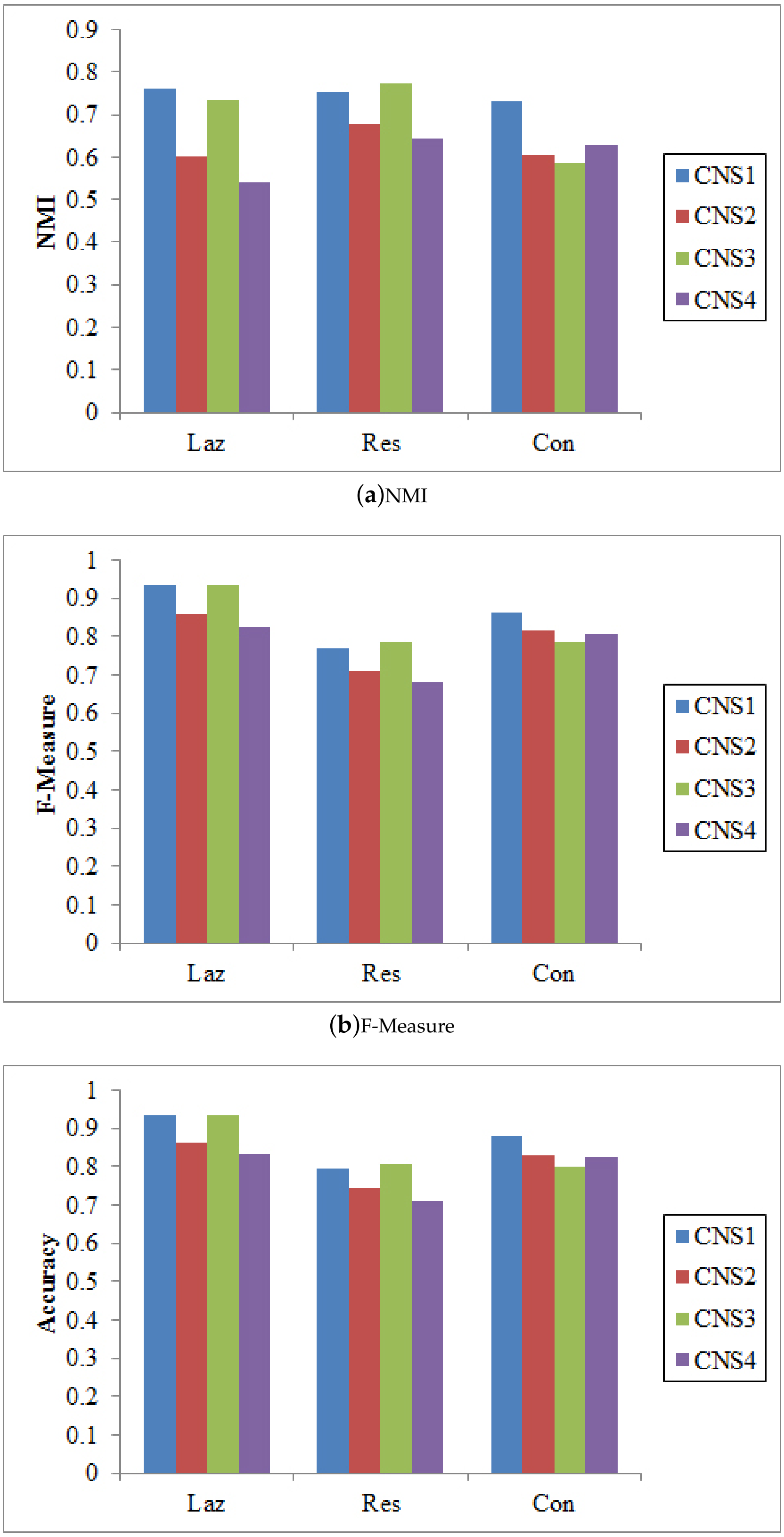


(**a**)NMI

(**b**)F-Measure

(**c**)Accuracy

Figure 4. The results of BGLL w.r.t. different CNSs.



(**a**)NMI

(**b**)F-Measure

(**c**)Accuracy

Figure 5. The results of Kmedoids w.r.t. different CNSs.

4.3. Impact of Varying Network Structure Complexity

We generate nine LFR benchmark networks with $N = 100$, $avgk = 5$, $maxk = 10$, $minc = 10$, and $maxc = 30$, but mu ranging from 0.1 to 0.9 to form networks with different structure complexities. Three attributes (Attributes 1, 2 and 3) are generated for these LFR networks according to the rules of synthetic node attributes and the noise level $nl = 0.3$.

Figure 6 reports the accuracy of the community detection results using BGLL on these networks. With the increase of mu, the level of separation between the communities decreases and the task of community detection is more difficult. Therefore, the accuracy of all methods decreases. However, CNS-based BGLL achieves better results than other similarity methods. Even when $mu = 0.9$, considering the complex interactions between node attributes and network structure still plays a positive role in the community detection process. BGLL based on Cosine and Jaccard similarity obtain almost the same results, with their two lines overlapping and are also the worst on all nine LFR networks. This verifies that simply considering common neighbors cannot accurately reveal their similarity.

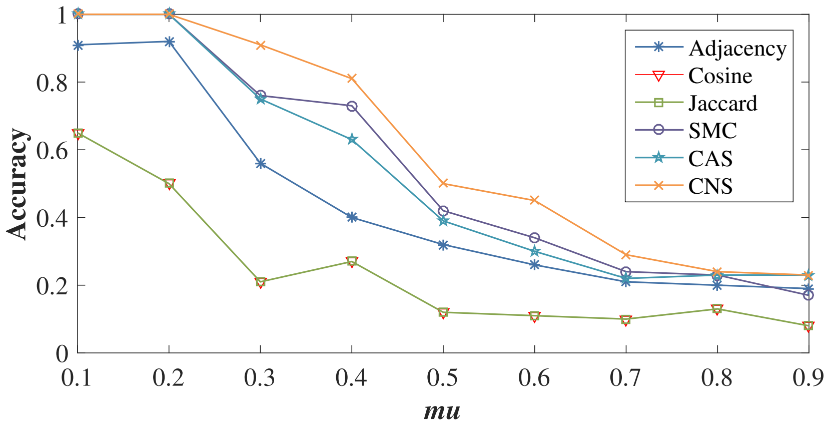


Figure 6. The results of BGLL w.r.t. different levels of network structure complexity.

4.4. Comparison Against Other Methods

We generate nine LFR benchmark networks with $N = 5000$, $avgk = 5$, $maxk = 10$, $minc = 10$, and $maxc = 30$, but mu ranging from 0.1 to 0.9 to form networks with different structure complexities. Three attributes (Attributes 1, 2 and 3) are generated for these LFR networks according to the rules of synthetic node attributes and the noise level $nl = 0.3$. We compare the results of CNS-based K-medoids with two other community detection algorithms on attributed networks, e.g., SA-cluster and CODICIL. The results are shown as Figure 7.

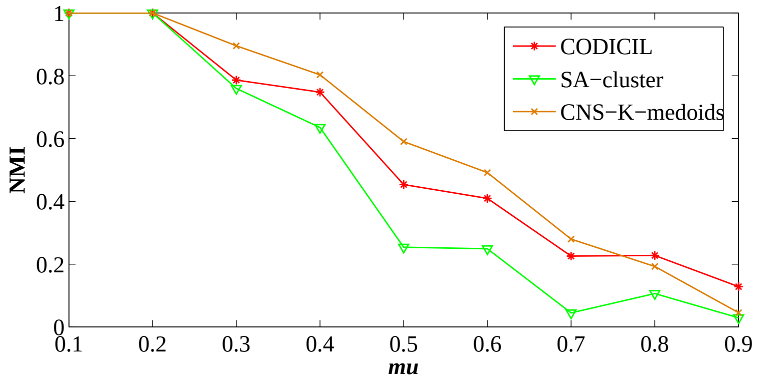


Figure 7. The results of NMI (%) w.r.t. three community detection algorithms on attributed networks.

From Figure 7, it is observed that the NMI of experimental results on nine different networks decreases with the increasing of parameter mu and the results of the proposed algorithm are optimal in most cases.

5. Conclusions

A novel coupled node similarity (CNS) measure is proposed to capture both explicit and implicit interactions between nodes using network structure and node attribute information in complex networks. Different levels of couplings in categorically attributed networks are learned, from node attribute values to nodes and between node attributes and network structure. Empirical analysis verifies the effectiveness of CNS-based community detection in beating several benchmark similarity methods, and, involving different node interactions and handling different levels of network structure complexity, highlights its strengths in terms of whether or not node attributes are involved. However, at present, our proposed method mainly concentrates on unweighted graphs. In the future, we will give some rules for the combination of the new and pre-existing weights to handle the weighted graphs. Our future work will also focus on using non-IID [36] learning on mixed attributed networks considering the coupling between different types of attributes at the attribute level.

Author Contributions: Conceptualization, F.M. and Z.W.; Methodology, F.M. and X.R.; Software, Y.X.; Validation, X.R., Y.X. and L.C.; Formal Analysis, X.R. and Z.W.; Investigation, X.R. and Y.X.; Resources, F.M.; Data Curation, Y.X. and L.C.; Writing—Original Draft Preparation, Y.X. and L.C.; Writing—Review & Editing, X.R. and Z.W.; Visualization, L.C.; Supervision, F.M.; Project Administration, Z.W.; Funding Acquisition, F.M.

Funding: This work was funded by the National Key Research and Development Program (No. 2016YFC060908) and the National Natural Science Foundation of China (No. U1610124, No. 71774159).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chai, B.F.; Yu, J.; Jia, C.Y.; Yang, T.B.; Jiang, Y.W. Combining a popularity-productivity stochastic block model with a discriminative-content model for general structure detection. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2013**, *88*, 012807. [CrossRef] [PubMed]
2. Xu, Z.; Ke, Y.; Wang, Y.; Cheng, H.; Cheng, J. GBAGC: A General Bayesian Framework for Attributed Graph Clustering. *ACM Trans. Knowl. Discov. Data* **2014**, *9*, 1–43. [CrossRef]
3. Yang, J.; Mcauley, J.; Leskovec, J. Community Detection in Networks with Node Attributes. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013; pp. 1151–1156.
4. Xin, Y.; Yang, J.; Xie, Z. A Semantic Overlapping Community Detection Algorithm in Social Networks Based on Random Walk. *J. Comput. Res. Dev.* **2015**, *52*, 499–511.
5. Cruz, J.D.; Bothorel, C.; Poulet, F. Entropy based community detection in augmented social networks. In Proceedings of the International Conference on Computational Aspects of Social Networks, Salamanca, Spain, 19–21 October 2011; pp. 163–168.
6. Ruan, Y.; Fuhry, D.; Parthasarathy, S. Efficient community detection in large networks using content and links. In Proceedings of the International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1089–1098.
7. Zhou, Y.; Cheng, H.; Yu, J.X. Graph Clustering based on Structural/Attribute Similarities. *Proc. VLDB Endow.* **2009**, *2*, 718–729. [CrossRef]
8. Zhou, Y.; Cheng, H.; Yu, J.X. Clustering large attributed graphs: An efficient incremental approach. In Proceedings of the International Conference on Data Mining, Sydney, Australia, 13–17 December 2010; pp. 689–698.
9. Cheng, H.; Zhou, Y.; Yu, J.X. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Trans. Knowl. Discov. Data* **2011**, *5*, 12. [CrossRef]
10. Steinhaeuser, K.; Chawla, N.V. Community Detection in a Large Real-World Social Network. In *Social Computing, Behavioral Modeling, and Prediction*; Springer: Boston, MA, USA, 2008; pp. 168–175.

11. Chanwimalueang, T.; Mandic, D. Cosine Similarity Entropy: Self-Correlation-Based Complexity Analysis of Dynamical Systems. *Entropy* **2017**, *19*, 652. [CrossRef]
12. Lee, S. Improving Jaccard Index for Measuring Similarity in Collaborative Filtering. In Proceedings of the International Conference on Information Science and Applications, Macau, China, 20–23 March 2017; pp. 799–806.
13. De Meo, P.; Ferrara, E.; Fiumara, G.; Provetti, A. Enhancing community detection using a network weighting strategy. *Inf. Sci. Int. J.* **2013**, *222*, 648–668.
14. Zhang, H.; Zhou, C.; Liang, X.; Zhao, X.; Li, Y. A Novel Edge Weighting Method to Enhance Network Community Detection. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, China, 9–12 October 2015; pp. 167–172.
15. Khadivi, A.; Hasler, M. A weighting scheme for enhancing community detection in networks. In Proceedings of the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010; pp. 1–4.
16. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*; Wiley: Hoboken, NJ, USA, 1990.
17. Wang, C.; Cao, L.; Wang, M.; Li, J.; Wei, W.; Ou, Y. Coupled Nominal Similarity in Unsupervised Learning. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; pp. 973–978.
18. Wang, C.; Dong, X.; Zhou, F.; Cao, L.; Chi, C.H. Coupled Attribute Similarity Learning on Categorical Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 781–797. [CrossRef] [PubMed]
19. Wang, C.; She, Z.; Cao, L. Coupled Clustering Ensemble: Incorporating Coupling Relationships both between Base Clusterings and Objects. In Proceedings of the IEEE 29th International Conference on Data Engineering, Brisbane, Australia, 8–12 April 2013; pp. 374–385.
20. Liu, C.; Cao, L. A Coupled k-nearest Neighbor Algorithm for Multi-label Classification. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Ho Chi Minh City, Vietnam, 19–22 May 2015; pp. 176–187.
21. Fu, B.; Xu, G.; Cao, L.; Wang, Z.; Wu, Z. Coupling Multiple Views of Relations for Recommendation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Ho Chi Minh City, Vietnam, 19–22 May 2015; pp. 732–743.
22. Cao, L. Coupling learning of complex interactions. *Inf. Process. Manag.* **2015**, *51*, 167–186. [CrossRef]
23. Bothorel, C.; Cruz, J.D.; Magnani, M.; Micenkova, B. Clustering Attributed Graphs: Models, Measures and Methods. *Netw. Sci.* **2015**, *3*, 408–444. [CrossRef]
24. Kim, K.; Altmann, J. Effect of homophily on network formation. *Commun. Nonlinear Sci. Numer. Simul.* **2017**, *44*, 482–494. [CrossRef]
25. McPherson, M.; SmithLovin, L.; Cook, J.M. Birds of a Feather: Homophily in Social Networks. *Ann. Rev. Sociol.* **2001**, *27*, 415–444. [CrossRef]
26. Xie, J.; Szymanski, B.K. Towards Linear Time Overlapping Community Detection in Social Networks. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kuala Lumpur, Malaysia, 29 May–1 June 2012; pp. 25–36.
27. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks. *J. Statist. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
28. Yu, D.; Liu, G.; Guo, M.; Liu, X. An improved K-medoids algorithm based on step increasing and optimizing medoids. *Expert Syst. Appl.* **2018**, *92*, 464–473. [CrossRef]
29. Raghavan, U.N.; Albert, R.; Kumara, S. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Phys. Rev. E* **2007**, *76*, 036106. [CrossRef] [PubMed]
30. Zhao, W.L.; Deng, C.H.; Ngo, C.W. k-means: A revisit. *Neurocomputing* **2018**, *291*, 195–206. [CrossRef]
31. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark Graphs for Testing Community Detection Algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef] [PubMed]
32. Lazega, E. *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*; Oxford University Press: Oxford, UK, 2001.
33. Cross, R.; Parker, A. *The Hidden Power of Social Networks*; Harvard Business School Press: Boston, MA, USA, 2004.
34. Sun, P. Weighting Links based on Edge Centrality for Community Detection. *Phys. A Stat. Mech. Its Appl.* **2014**, *394*, 346–357. [CrossRef]

35. Hand, D.; Christen, P. A note on using the F-measure for evaluating record linkage algorithms. *Stat. Comput.* **2018**, *28*, 539–547. [CrossRef]

36. Cao, L. Non-IIDness Learning in Behavioral and Social Data. *Comput. J.* **2014**, *57*, 1358–1370. [CrossRef]