A framework for community detection in heterogeneous multi-relational networks

Xin Liu [1,2,3], Weichu Liu [1], Tsuyoshi Murata [1], and Ken Wakita [1,2]

[1]Tokyo Institute of Technology
2-12-1 Ookayama, Meguro, Tokyo, 152-8552 Japan

[2]CREST JST
K's Gobancho, 7, Gobancho, Chiyoda, Tokyo, 102-0076 Japan

[3]Wuhan University of Technology
122 Luoshi Road, Wuhan, Hubei, 430070 China

E-mail: {tsinllew@ai.cs, liu.w@ai.cs, murata@cs,
wakita@is}.titech.ac.jp

July 21, 2014

Abstract

There has been a surge of interest in community detection in homogeneous single-relational networks which contain only one type of nodes and edges. However, many real-world systems are naturally described as heterogeneous multi-relational networks which contain multiple types of nodes and edges. In this paper, we propose a new method for detecting communities in such networks. Our method is based on optimizing the composite modularity, which is a new modularity proposed for evaluating partitions of a heterogeneous multi-relational network into communities. Our method is parameter-free, scalable, and suitable for various networks with general structure. We demonstrate that it outperforms the state-of-the-art techniques in detecting preplanted communities in synthetic networks. Applied to a real-world Digg network, it successfully detects meaningful communities.

1 Introduction

Many social, biological, and information systems can be described as networks, where nodes represent fundamental entities of the system, such as individuals, users, genes, web pages, and so on; and edges represent relations or interactions between the entities. In recent years, there has been a surge of interest in analysis of networks. A highly discussed topic is community detection [16] — the identification of groups of closely related nodes that correspond to functional subunits of the underlying system, such as collections of pages on closely related topics on the web or groups of people with common interests in social media. Thus community detection provides insight into how the system is internally organized.

Previous studies on community detection mainly focus on homogeneous single-relational networks which contain only one type of nodes and edges. Many real-world systems, however, are naturally described as *heterogeneous* (contain multiple types of nodes) *multi-relational* (contain multiple types of edges) networks. Take the photo service website Flickr as an example. Flickr users can upload photos, annotate photos with tags, and establish friendship with other users. As shown in Fig. 1, Flickr can be described as a heterogeneous multi-relational network, which contains three types of nodes (users, photos, and tags) and three types of edges representing the above relations.

Traditionally we first simplify a heterogeneous multi-relational network to a homogeneous single-relational network and then conduct community analysis. Take the above Flickr network as an example. We can omit information about photos and tags, and detect communities directly in the homogeneous user friendship network. However, we may omit too much valuable information in this simplification process. For example, the data about users can be incomplete and noisy, and some active users have thousands of friends while some have no friends at all. Consequently, we cannot obtain real user communities based on the friendship network alone. In this scenario, we expect that a method which effectively utilizes multi-faceted information in the heterogeneous multi-relational network would enhance the community detection results.

In this paper, we propose a new method for detecting communities in heterogeneous multi-relational networks. Our method follows the line of the modularity optimization method which is widely used for detecting communities in homogeneous single-relational networks [36, 8, 37, 4, 12, 31, 53, 43, 44, 3, 26, 46, 19]. In specific, we first propose a *composite modularity* for evaluating the "goodness" of a partition of a heterogeneous multi-relational

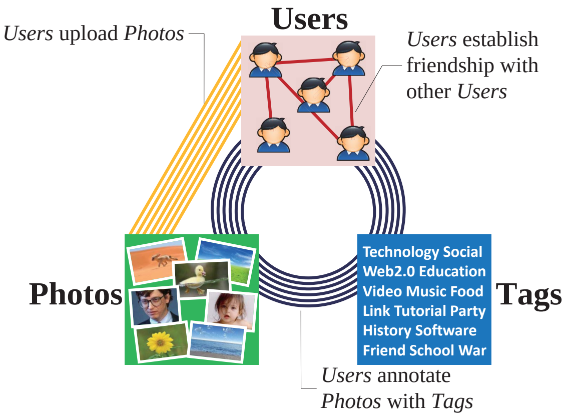
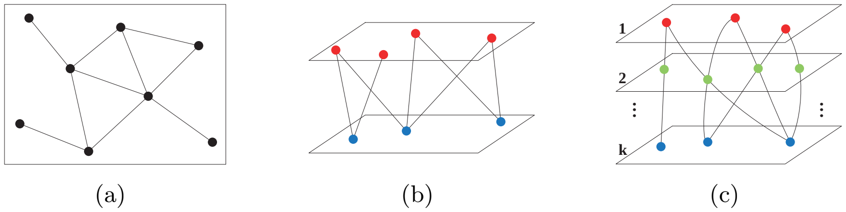


Figure 1: Describing Flickr as a heterogeneous multi-relational network. This network contains three types of nodes: 1) users, 2) photos, 3) tags; and three types of edges: 1) the edges representing friendship between users, 2) the edges representing that users unload photos, 3) the 3-way hyper-edges representing that users annotate photos with tags.

network into communities. Then we develop a fast algorithm for optimizing the composite modularity and detecting the communities. Our method is consistent with the modularity optimization method, because the composite modularity reduces to modularity when we deal with a homogeneous single-relation network. Our method is applicable to networks of general structure, which may even contain hyper-edges for representing relations between more than two nodes. Another advantage of our method is that it is parameter-free and can automatically detect communities without any a priori knowledge such as the number of communities. Experiments in synthetic networks show that our method outperforms the state-of-the-art techniques in detecting pre-planted communities. We use a real-world Digg network to illustrate that our method successfully detects meaningful communities.

The rest of the paper is organized as follows. Section 2 reviews related research. Section 3 formulates the problem of community detection in a general heterogeneous multi-relational network. Section 4 introduces our new method. Section 5 presents experimental results, followed by a conclusion in Section 6.



(a) (b) (c)

Figure 2: (a) A unipartite network. (b) A bipartite network. (c) A k-partite network.

2 Related Work

The study of community detection in homogeneous single-relational networks or called *unipartite networks* (see Fig. 2(a)) has a long history. It is closely related to graph partitioning [20] in computer science, and hierarchical clustering [45] in sociology. In the past decade, this study has attracted a great deal of interest and various methods were proposed [13, 38, 10, 22, 40]. In particular, a family of methods which are widely used is known as *modularity optimization*. Modularity was originally proposed by Newman and Girvan [16] for evaluating the "goodness" of a partition of a unipartite network into communities. The definition of modularity involves a comparison of the fraction of intra-community edges in the observed network minus the expected value of that fraction in a randomized network, which is called the null model. More precisely, the mathematical expression of modularity in an undirected single-relational network reads

$$\mathrm{Q}(\mathcal{L}) = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1}^{n} (A_{ij} - P_{ij})\, \delta(l_i, l_j), \tag{1}$$

where n is the number of nodes, m is the number of edges, $\mathcal{L} = \{l_1, l_2, \cdots, l_n\}$ is a partition, with element l_i indicating the community membership of the i-th node v_i, A_{ij} is the number of edges between v_i and v_j in the observed network, P_{ij} is the expected value of that number in the null model, and δ is the Kronecker's delta. In addition, another equivalent expression of modularity is

$$Q(\mathcal{L}) = \sum_{\alpha=1}^{c} (e_{\alpha\alpha} - a_{\alpha}^{2}), \tag{2}$$

where

$$e_{\alpha\alpha} = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \delta(l_i, \alpha) \delta(l_j, \alpha) \tag{3}$$

is the fraction of edges within community V_α, and

$$a_\alpha = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \delta(l_i, \alpha) \tag{4}$$

is the fraction of edges attached to community V_α.

Optimizing modularity is proved to be NP-hard [6]. Researchers have developed various heuristic optimization algorithms [36, 8, 37, 4, 12, 31, 53, 43, 44, 3, 26, 46, 19]. In particular, the simulated annealing algorithm [31] is the most accurate (in terms of the modularity score) [10]. However, this algorithm requires a long time to complete, and is only suitable for small-scale networks. On the other hand, the label propagation algorithm [3], which requires only near linear time to complete, is perhaps the fastest. However, this algorithm tends to get stuck in a poor local optimum [26]. In practice, the Louvain algorithm [4] is widely used, since it reaches a proper balance between accuracy and speed.

A notable issue of modularity optimization is the resolution limit, which refers to the incapability of detecting small communities in large-scale networks [14, 17]. Researchers have tried to get around this issue by proposing variants of modularity. For example, Arenas et al. modified modularity by adding a parameter that forms self-loop for each node [1]. Reichardt and Bornholdt modified modularity by adding a parameter in front of the null model term [42]. Both parameters can be used to control the resolution level and detect communities at multiple resolutions. However, a recent study by Lancichinetti and Fortunato demonstrated that these methods are intrinsically deficient and still suffer from the resolution limit [23].

There are studies on community detection in heterogeneous single-relational networks. The most simple model of such networks is the bipartite network, where there are two types of nodes and edges that connect nodes of different types (see Fig. 2(b)). Examples of bipartite networks include author-paper networks, actor-movie networks, and customer-product networks. A direct generalization of bipartite network is the k-partite network, where there are k types of nodes and hyper-edges that connect k nodes of different types (see Fig. 2(c)). Researchers extended modularity to bipartite networks and k-partite networks. For example, Guimerà et al. proposed a bipartite modularity which focuses on evaluating the partition of only one type of nodes, and used simulated annealing algorithm for optimization [18]. Barber proposed a bipartite modularity which assumes a bipartite structure of the null model, and devised a label propagation algorithm called BRIM [2] for optimization. Murata proposed a k-partite modularity in a unified way as the definition of modularity, in the sense that his k-partite modularity can reduce to modularity if the k-partite network becomes a unipartite network [34, 33]. Neubauer et al. proposed another k-partite modularity by reducing a k-partite network to $\binom{k}{2}$ bipartite networks and utilizing Murata's definition of bipartite modularity [35]. Murata and Neubauer et al. employed greedy bottom-up algorithm for optimization. In addition to the family of methods based on optimizing variants of modularity, Sun et al. and Liu et al., respectively, proposed information compression based methods for bipartite networks [47] and k-partite networks [27]. Moreover, there are methods for simultaneously clustering related sets of heterogeneous data, such as documents and words. Such methods are often referred to co-clustering [7, 11, 30].

There are studies on community detection in homogeneous multi-relational networks (sometimes called the multi-mode networks, multi-dimensional networks, or multi-slice networks). For example, researchers developed methods for detecting communities in a particular subclass of such networks, known as signed networks where each edge has a positive or negative sign [52, 54, 5]. Mucha et al. proposed a multiplex model for describing a homogeneous multi-relational network and developed a method based on optimizing a generalized modularity known as stability [32]. Moreover, researchers proposed methods based on matrix approximation [50] and spectral analysis [51].

Recent research have also addressed the problem of community detection in heterogeneous multi-relational networks. Comar et at. developed a method based on non-negative matrix factorization [9]. However, their work is restricted to a specific subclass of networks which contains two types of nodes and three types of edges. Sun et al. designed a ranking-based community detection method for a specific subclass of networks, referred to the star network schema [49]. Thus both of these two methods are not applicable to general networks with any possible structure. As for the problem in a general heterogeneous multi-relational network, a naive approach is to simplify the network to a single-relational network and then conduct community detection. However, valuable information might be omitted, leading to inaccurate results. Popescul et al. proposed a method by calculating the similarity between nodes and building a similarity matrix [41]. However, when the structure of a network becomes complex, it is difficult to find a

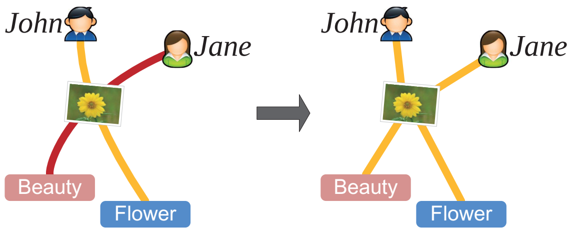
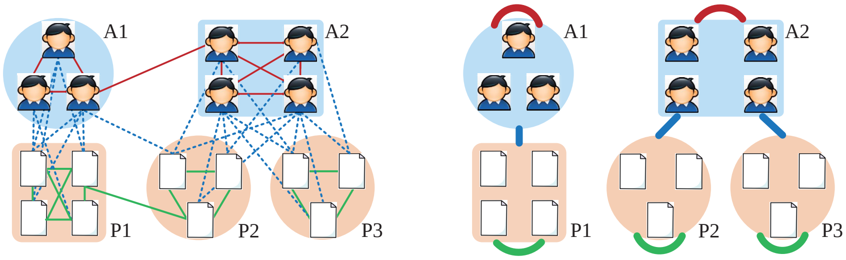


Figure 3: Reducing hyper-edges to normal edges causes information loss. The curved lines indicate 3-way hyper-edges.

reasonable similarity measure. Also, high computational complexity is another issue, which prevents this method from being applied to large-scale networks. In addition, Lin et al. proposed a method based on tensor factorization [25]. Different from our work, they suppose that a community contains nodes of different types. If we partition nodes of different types separately, it means that nodes of different types have the same number of communities. Unfortunately, this situation is rarely seen in real-world scenario. Another important work is Ref. [48], where the authors considered community detection in networks with incomplete attributes. The difference from our work is that their method utilizes both edges and node attributes. Note that a drawback of the above methods is that they require a priori knowledge about the number of communities. This limits their usage in inferring the latent organization of a real system.

Hyper-edges are seldom discussed in previous studies. In general, hyper-edges are useful for representing relations that involve more than two nodes. For example, in a social tagging system such as Flickr, users use tags to annotate photos. Such a relation can be naturally represented as a 3-way hyper-edge. Although we can reduce the 3-way hyper-edge to a normal edge, some information is lost during the reduction process. As shown in Fig. 3, *John* annotates the photo with *Flower*, and *Jane* annotates the photo with *Beauty*. If we reduce the hyper-edges to normal edges, we cannot distinguish who uses *Flower* and who uses *Beauty*. In this paper, we aim at automatically detecting communities in a general heterogeneous multi-relational network which may contain hyper-edges, without a priori knowledge about the number of communities.



(a) (b)

Figure 4: (a) The link pattern based community. (b) The link patterns of the communities.

3 Problem Formulation

In homogeneous single-relational networks, people mainly focus on densely intra-connected and sparsely inter-connected community. In heterogeneous multi-relational networks, this concept can be generalized to the link pattern based community [28, 29] — a group of nodes that have the similar link patterns, i.e., the nodes within a community connect to other nodes in similar ways. Fig. 4(a) shows a heterogeneous multi-relational network with two types of nodes (author and paper nodes), and three types of edges (the edges representing the friendship between authors, the authorship between authors and papers, and the citation relationship between papers). This network has two author communities (A1 and A2), and three paper communities (P1, P2, and P3). Take the community A1 as an example. The nodes in A1 have the similar link patterns, as they all densely connect to the nodes in A1 and P1, and sparsely connect to the nodes in A2, P2, and P3. Similar interpretation applies to other communities. Fig. 4(b) shows the link patterns of these communities. Note that the definition of link pattern based community is reasonable, as the nodes with similar link patterns are likely to share common features and form the real community.

In the following, we formulate the problem of community detection in a general heterogeneous multi-relational network. Now suppose a heterogeneous multi-relational network $\mathbf{G} = (\mathbf{V}^{[1]} \cup \mathbf{V}^{[2]} \cup \cdots \cup \mathbf{V}^{[r]}, \mathbf{E}^{[1]} \cup \mathbf{E}^{[2]} \cup \cdots \cup \mathbf{E}^{[s]})$, where there are r types of nodes and s types of edges. $\mathbf{V}^{[x]}$ is the node set of the x-th type. $\mathbf{E}^{[y]}$ is the edge set of the y-th type. $\mathbf{E}^{[y]}$ should satisfy either of the following two conditions:

1. There exists a $x \in \{1, 2, \cdots, r\}$, such that $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x]} \times \mathbf{V}^{[x]}$, i.e., $\mathbf{E}^{[y]}$ is a set of edges that connect nodes of the same type.

Table 1: Notations for a heterogeneous multi-relational network $\mathbf{G}$

Symbol	Meaning
n	The total number of nodes
m	The total number of edges
r	The number of node types
s	The number of edge types
$\mathbf{V}^{[x]}$	The node set of the x-th type
$\mathbf{E}^{[y]}$	The edge set of the y-th type
$\mathbf{G}^{[y]}$	The subnetwork consisting of $\mathbf{E}^{[y]}$ and the incident nodes
$\mathbf{A}^{[y]}$	The connectivity array of $\mathbf{G}^{[y]}$
$n^{[x]}$	The number of nodes in $\mathbf{V}^{[x]}$
$c^{[x]}$	The number of communities in $\mathbf{V}^{[x]}$
$m^{[y]}$	The number of edges in $\mathbf{E}^{[y]}$
$v_i^{[x]}$	The i-th node in $\mathbf{V}^{[x]}$
$l_i^{[x]}$	The community membership of $v_i^{[x]}$
$V_\alpha^{[x]}$	The α-th community in $\mathbf{V}^{[x]}$

2. There exists $x_1, x_2, \cdots, x_k \in \{1, 2, \cdots, r\}$ $(k \leq r)$ which are not equal to each other, such that $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x_1]} \times \mathbf{V}^{[x_2]} \times \cdots \times \mathbf{V}^{[x_k]}$, i.e., $\mathbf{E}^{[y]}$ is a set of k-way edges * that connect nodes of different types.

Given $\mathbf{G}$, the problem is to find a "good" partition $\mathcal{L} = \mathcal{L}^{[1]} \cup \mathcal{L}^{[2]} \cup \cdots \cup \mathcal{L}^{[r]}$, such that $\mathcal{L}^{[x]}$ divides $\mathbf{V}^{[x]}$ into disjoint communities $(x = 1, \cdots, r)$. The meaning of "good" is that the nodes in each community have the similar link patterns. Note that the number of communities in each node set is not known a priori.

4 Method Based on Optimizing the Composite Modularity

In this section, we propose our method for the problem formulated in Section 3. Our method follows the line of the modularity optimization method which is widely used for detecting communities in homogeneous single-relational networks. The idea is to define a quality function modularity for evaluating the goodness of community partitions, and then search a partition with a high modularity score. Inspired by this idea, we propose a new quality function — the composite modularity for evaluating the goodness of a partition of a heterogeneous multi-relational network into communities, and develop a fast algorithm for optimizing the composite modularity. In

*If $k > 2$ the edges are actually hyper-edges.

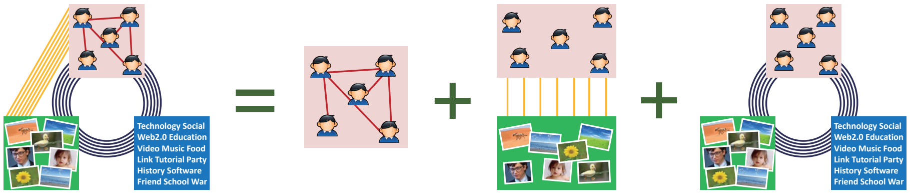


Figure 5: The Flickr heterogeneous multi-relational network can be regarded as an integration of a unipartite subnetwork, a bipartite subnetwork, and a tripartite subnetwork.

the following, we first propose the composite modularity, and then present its optimization algorithm. For convenience, we list the major notations in Table 1.

4.1 The Composite Modularity

A heterogeneous multi-relational network $\mathbf{G} = (\mathbf{V}^{[1]} \cup \mathbf{V}^{[2]} \cup \cdots \cup \mathbf{V}^{[r]}, \mathbf{E}^{[1]} \cup \mathbf{E}^{[2]} \cup \cdots \cup \mathbf{E}^{[s]})$ contains multiple types of nodes and edges. In another way, we can take $\mathbf{G}$ as an integration of multiple subnetworks. For example, we can take the Flickr heterogeneous multi-relational network as an integration of three subnetworks, as illustrated in Fig. 5. Suppose $\mathbf{G}^{[y]}$ denotes the subnetwork which consists of $\mathbf{E}^{[y]}$ and the incident nodes. We can represent $\mathbf{G}$ as $\mathbf{G} = \mathbf{G}^{[1]} \cup \mathbf{G}^{[2]} \cup \cdots \cup \mathbf{G}^{[s]}$.

For each subnetwork, either a unipartite subnetwork or a k-partite subnetwork, researchers have proposed a modularity for evaluating partition of the related node sets [39, 34, 33]. We can integrate these modularities and define a composite modularity for evaluating partition of all node sets as follows

$$Q(\mathcal{L}) = \sum_{y=1}^{s} \frac{m^{[y]}}{m} Q^{[y]}(\mathcal{L}). \tag{5}$$

Here $m^{[y]} = |\mathbf{E}^{[y]}|$ is the number of edges in $\mathbf{G}^{[y]}$, $m = \sum_{y=1}^{s} m^{[y]}$ is the total number of edges, and $Q^{[y]}$ is the modularity in $\mathbf{G}^{[y]}$.

For a unipartite subnetwork $\mathbf{G}^{[y]}$, $Q^{[y]}$ is the modularity proposed by Newman and Girvan [39]. For convenience of description, we suppose all nodes of $\mathbf{V}^{[x]}$ are contained in $\mathbf{G}^{[y]}$, or mathematically, $\mathbf{G}^{[y]} = (\mathbf{V}^{[x]}, \mathbf{E}^{[y]})$. The partition on $\mathbf{V}^{[x]}$ is $\mathcal{L}^{[x]} = \{V_\alpha^{[x]} \,|\, \alpha = 1, \cdots, c^{[x]}\}$ *, where $V_\alpha^{[x]}$ denotes

*In the case that not all nodes of $\mathbf{V}^{[x]}$ are contained in $\mathbf{G}^{[y]}$, for example, $\mathbf{G}^{[y]} =$

the α-th community in $\mathbf{V}^{[x]}$, and $c^{[x]}$ denotes the number of communities in $\mathbf{V}^{[x]}$. Then

$$Q^{[y]}(\mathcal{L}) = Q^{[y]}(\mathcal{L}^{[x]}) = \sum_{\alpha=1}^{c^{[x]}} (e_{\alpha\alpha}^{[y]} - {a_{\alpha}^{[y]}}^2), \tag{6}$$

where

$$e_{\alpha\alpha}^{[y]} = \frac{1}{2m^{[y]}} \sum_{i=1}^{n^{[x]}} \sum_{j=1}^{n^{[x]}} A_{ij}^{[y]} \delta(l_i^{[x]}, \alpha) \delta(l_j^{[x]}, \alpha) \tag{7}$$

is the fraction of subnetwork edges within community $V_{\alpha}^{[x]}$, and

$$a_{\alpha}^{[y]} = \frac{1}{2m^{[y]}} \sum_{i=1}^{n^{[x]}} \sum_{j=1}^{n^{[x]}} A_{ij}^{[y]} \delta(l_i^{[x]}, \alpha) \tag{8}$$

is the fraction of subnetwork edges attached to community $V_{\alpha}^{[x]}$.

For a k-partite subnetwork $\mathbf{G}^{[y]}$, $Q^{[y]}$ is a variant k-partite modularity proposed by us previously [34, 33]. For convenience of description, we also suppose all nodes of $\mathbf{V}^{[x_1]}, \mathbf{V}^{[x_2]}, \cdots, \mathbf{V}^{[x_k]}$ are contained in $\mathbf{G}^{[y]}$, or mathematically, $\mathbf{G}^{[y]} = (\mathbf{V}^{[x_1]} \cup \mathbf{V}^{[x_2]} \cup \cdots \cup \mathbf{V}^{[x_k]}, \mathbf{E}^{[y]})$. The partition on $\mathbf{V}^{[x_1]}, \mathbf{V}^{[x_2]}, \cdots, \mathbf{V}^{[x_k]}$ is $\mathcal{L}^{[x_1]} \cup \mathcal{L}^{[x_2]} \cup \cdots \cup \mathcal{L}^{[x_k]} = \{V_{\alpha_1}^{[x_1]} \,|\, \alpha_1 = 1, \cdots, c^{[x_1]}\} \cup \{V_{\alpha_2}^{[x_2]} \,|\, \alpha_2 = 1, \cdots, c^{[x_2]}\} \cup \cdots \cup \{V_{\alpha_k}^{[x_k]} \,|\, \alpha_k = 1, \cdots, c^{[x_k]}\}$ *. Then,

$$\begin{aligned} Q^{[y]}(\mathcal{L}) &= Q^{[y]}(\mathcal{L}^{[x_1]} \cup \mathcal{L}^{[x_2]} \cup \cdots \cup \mathcal{L}^{[x_k]}) & (9) \\ &= \frac{1}{k} \sum_{t=1}^{k} \sum_{\alpha_t=1}^{c^{[x_t]}} (e_{\alpha_1 \cdots \alpha_k}^{[y]} - a_{\alpha_1}^{[x_1,y]} \cdots a_{\alpha_k}^{[x_k,y]}) & (10) \end{aligned}$$

where

$$\alpha_1, \cdots, \alpha_{t-1}, \alpha_{t+1}, \cdots, \alpha_k = \underset{\tilde{\alpha_1}, \cdots, \tilde{\alpha_{t-1}}, \tilde{\alpha_{t+1}}, \cdots, \tilde{\alpha_k}}{\arg\max} e_{\tilde{\alpha_1} \text{-} \tilde{\alpha_{t-1}}, \alpha_t \tilde{\alpha_{t+1}} \text{-} \tilde{\alpha_k}}^{[y]} \tag{11}$$

$(\mathbf{V}'^{[x]}, \mathbf{E}^{[y]})$, where $\mathbf{V}'^{[x]} \subset \mathbf{V}^{[x]}$. Then the partition on $\mathbf{V}'^{[x]}$ is $\mathcal{L}'^{[x]} = \{V_{\alpha}^{[x]} \cap \mathbf{V}'^{[x]} \,|\, V_{\alpha}^{[x]} \cap \mathbf{V}'^{[x]} \neq \emptyset, \alpha = 1, \cdots, c^{[x]}\}$.

*In the case that not all nodes of $\mathbf{V}^{[x_1]}, \mathbf{V}^{[x_2]}, \cdots, \mathbf{V}^{[x_k]}$ are contained in $\mathbf{G}^{[y]}$, please refer to the solution for a similar case in the unipartite subnetwork.

indicate the indices of the communities which have the largest number of subnetwork edges to community $V_{\alpha_t}^{[x_t]}$,

$$e_{\alpha_1\text{-}\alpha_k}^{[y]} = \frac{1}{m^{[y]}} \sum_{i_1=1}^{n^{[x_1]}} \cdots \sum_{i_k=1}^{n^{[x_k]}} A_{i_1 \cdots i_k}^{[y]} \delta(l_{i_1}^{[x_1]}, \alpha_1) \cdots \delta(l_{i_k}^{[x_k]}, \alpha_k) \tag{12}$$

is the fraction of subnetwork edges between community $V_{\alpha_1}^{[x_1]}, V_{\alpha_2}^{[x_2]}, \cdots, V_{\alpha_k}^{[x_k]}$, and

$$a_{\alpha_t}^{[x_t,y]} = \frac{1}{m^{[y]}} \sum_{i_1=1}^{n^{[x_1]}} \cdots \sum_{i_k=1}^{n^{[x_k]}} A_{i_1 \cdots i_k}^{[y]} \delta(l_{i_t}^{[x_t]}, \alpha_t) \tag{13}$$

is the fraction of subnetwork edges attached to community $V_{\alpha_t}^{[x_t]}$.

In general Q ranges in $[-1, 1]$, as each $Q^{[y]}$ ranges in $[-1, 1]$. Note that the composite modularity is consistent with Newman and Girvan's modularity [16] and our k-partite modularity [34, 33] — If the heterogeneous multi-relational network reduces to a unipartite network/k-partite network, the composite modularity recovers Newman and Girvan's modularity/our k-partite modularity.

In the above definition of the composite modularity, the component modularities are weighted by the fraction of edges of the subnetwork. An underlying assumption of this weighting strategy is that edges are treated equally, no matter their types. This is a natural assumption in the case where we have no background knowledge about the network. However, in the case that some type of edges account for a dominant proportion of the total number of edges, or some type of edges contain much noise and should be attached less weight, the above definition may be limited in detecting the communities. In these cases, we should use other weighting strategies based on some a priori knowledge of the network. For example, we can define the composite modularity as

$$Q(\mathcal{L}) = \sum_{y=1}^{s} \frac{1}{w^{[y]}} Q^{[y]}(\mathcal{L}), \tag{14}$$

where $w^{[y]}$ represents the relative importance of the y-th type edges, and is specified by the experimenter.

4.2 Optimization Algorithm

Optimizing the composite modularity is NP-hard [6]. We develop an efficient heuristic algorithm for practical use. Our algorithm is based on the Louvain algorithm, originally designed by Blondel et al. for optimizing Newman and Girvan's modularity [4]. In the following, we first describe how to optimize the composite modularity by the Louvain algorithm, and then discuss how this algorithm can be improved for practical application.

Louvain algorithm contains two phases which are performed iteratively until maximal Q is reached. In the first phase, each node is initially assigned to a community of itself. Then, for each node $v_i^{[x]}$ ($x \in \{1, 2, \cdots, r\}$, $i \in \{1, 2, \cdots, n^{[x]}\}$), the gains in Q that would result from moving $v_i^{[x]}$ to the community of each of its counterparts ($v_i^{[x]}$'s 1-hop and 2-hop neighbors which are of the same node type as $v_i^{[x]}$) are calculated, and $v_i^{[x]}$ is moved to the community for which the maximum positive gain in Q is attained. If no positive gain is possible, $v_i^{[x]}$ stays in its original community. This subprocess is repeated sequentially for all nodes until no individual move will result in a gain, marking the end of the first phase. In the second phase, a new network is built whose nodes are the communities in the first phase. The weights of the edges between the new nodes are calculated as the sum of the weights of the edges between nodes in the corresponding communities. Once this second phase is completed, the two phases are repeated iteratively until there are no changes.

The efficiency of the first phase relies on calculation of the gains in Q resulting from moving $v_i^{[x]}$ to the community of each of its counterparts. This operation requires a computational complexity of $O(\bar{d}^{2k_{max}})$ in the worst case, where $\bar{d}$ is the average node degree and k_{max} is the highest partite number for the k-partite subnetworks. Thus the complexity of the first phase is $O(\bar{d}^{2k_{max}})$. The complexity of the second phase is $O(n)$. In practice, the number of iterations of the two phases is generally small [4]. Consequently, the total computational complexity of Louvain algorithm for optimizing the composite modularity is near $O(\bar{d}^{2k_{max}} \cdot n)$. This complexity may prohibit practical implementations for large-scale networks which contain millions of nodes.

To speed up the algorithm, we propose a Louvain-C algorithm with the strategy of *divide and rule*. Given a heterogeneous multi-relational network which is an integration of multiple subnetworks, Louvain-C algorithm consists of the following three steps.

(1) Detect communities in each subnetwork seperately;

(2) Combine the partitions obtained in each subnetwork and derive some constraints;

Algorithm 1 Find the *must-be-assigned-together* constraints

```
Input: The community partitions for each subnetwork (each community
       has a global unique ID)
Output: Sets of nodes which are assigned to the same community by every
        related partition
begin
    Iterate the partitions for each subnetwork and build a list for each node
    $v_i^{[x]}$. The list records the IDs of the communities to which $v_i^{[x]}$ belongs
    (referred as $v_i^{[x]}$'s community trace).
    // Nodes assigned to the same community by every related
       partition should have the same community trace, and vice
       versa.
    Examine the above records and build a map, where a key is a community
    trace and its value is the nodes which have this community trace.
    Then each value represents a set of nodes which must be assigned to-
    gether. Return these sets as output.
end
```

(3) Optimize the composite modularity under these constraints.

As for step (1), we can use modularity and k-partite modularity optimization methods to detect communities in each subnetwork. Other existing methods (see Section 2) can also be used. Since a node can be involved in multiple subnetworks, it may be assigned to different communities by partitions in different subnetworks. Thus, we combine these partitions and derive some consistent constraints in step (2). Nodes $v_{i_1}^{[x]}$, $v_{i_2}^{[x]}$, $\cdots$, $v_{i_k}^{[x]}$ form a *must-be-assigned-together* constraint if they are assigned to the same community by every related partition. Algorithm 1 shows the procedure of finding such constraints. The computational complexity of Algorithm 1 is $O(n{\cdot}logn)$. In step (3), we first build a new network based on these constraints, in a similar way as the second phase of Louvain algorithm. In other words, a node in the new network corresponds to a group of nodes that must be assigned together and weights between the new nodes are recalculated. Then, we optimize the composite modularity in this new network. Note that the size of the new network is much smaller than the original one. On the other hand, building the new network has little impact on the optimization result, since we just aggregate the nodes that "must be assigned together". As a result, Louvain-C algorithm can dramatically reduce the computational

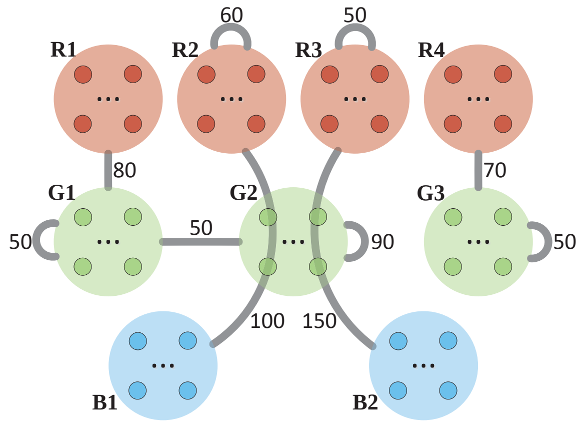


Figure 6: The link patterns of the communities in the synthetic network. The numbers indicate the numbers of edges.

time without compromising the accuracy. Please see the next section for more information about comparison of the two algorithms.

5 Experiments

So far we have proposed the composite modularity optimization method for detecting communities in heterogeneous multi-relational networks. In this section, we present experiments for testing performance of our method.

5.1 Comparison with Other Methods in Synthetic Networks

First, we compare our method with others in synthetic networks. The basic scheme is as follows. 1) We generate a sequence of synthetic heterogeneous multi-relational networks with planted communities. 2) Applying various methods to these networks (the planted partition is hidden at this time), we test which method can detect most of the planted communities. Such kind of testing is widely used by other researchers in the community detection field [10, 16, 24, 21, 22].

As shown in Fig. 6, our synthetic network model contains three types of nodes (the red, green, and blue nodes), and four types of edges (the edges between red nodes, the edges between green nodes, the edges between red and green nodes, and the hyper-edges between red, green, and blue nodes). Red nodes are organized into four communities, each containing 15 nodes. Green nodes are organized into three communities, each containing 20 nodes.

Blue nodes are organized into two communities, each containing 25 nodes. From Fig. 6 we can see that each community has its own representative link pattern (the number of edges are shown in the figure). For example, the link pattern of community G1 is that its nodes all densely connect to nodes in G1 (edge density = $\frac{50}{\text{Size(G1)}\times[\text{Size(G1)}-1]/2} = 0.2632$), to nodes in R1 (edge density = $\frac{80}{\text{Size(G1)}\times\text{Size(R1)}} = 0.2667$), and to nodes in G2 (edge density = $\frac{50}{\text{Size(G1)}\times\text{Size(G1)}} = 0.1250$). Based on these link patterns, we generated a total of 750 edges. Then, we add noise edges randomly. The noise rate increases from 0% to 200% (thus the total number of edges ranges from 750 to 2,250), so that the planted communities are more and more difficult to be detected.

To evaluate a method, we apply it to the network and calculate the similarity between the obtained partition and the planted partition. The more similar the two partitions, the better the method. We adopted the regularly used *normalized mutual information* (NMI) [15, 10] to quantify the similarity between two partitions $\mathcal{L}_1$ and $\mathcal{L}_2$. If $\mathcal{L}_1$ and $\mathcal{L}_2$ match completely, we have a maximum NMI value of 1, whereas if $\mathcal{L}_1$ and $\mathcal{L}_2$ are totally independent of one another, we have a minimum value of 0.

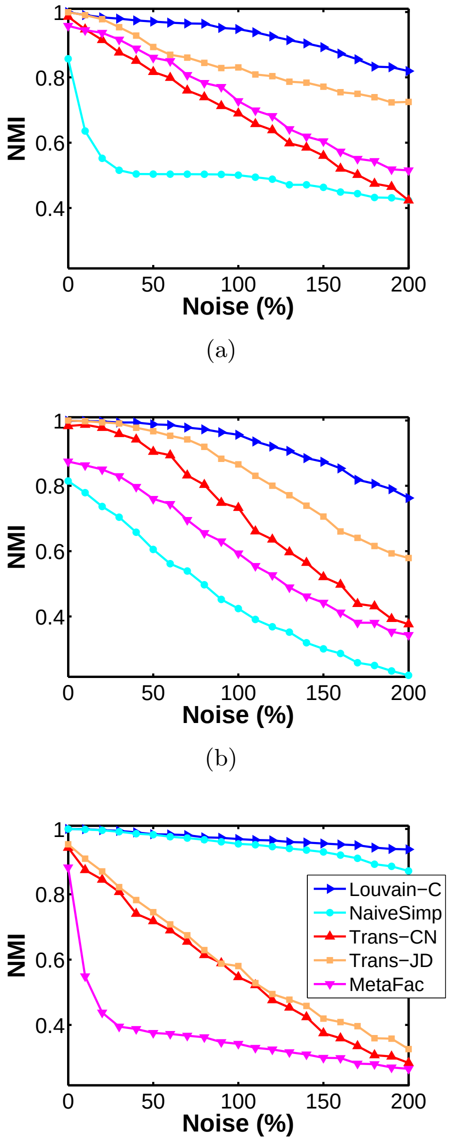
We compare our method with the following four methods, which cover the state-of-the-art techniques.

- *NaiveSimp*: Simplify the heterogeneous multi-relational network to a single-relational network and detect communities (the results are based on the best performance obtained in the single-relational network for each edge type).

- *Trans-CN* (A modification of the method proposed in Ref. [41]): Transform the heterogeneous multi-relational network to weighted unipartite networks for each node type based on the number of common neighbors. Then detect communities in each unipartitie network separately.

- *Trans-JD* (A modification of the method proposed in Ref. [41]): Transform the heterogeneous multi-relational network to weighted unipartite networks for each node type based on the Jaccard index. Then detect communities in each unipartitie network separately.

- *MetaFac* [25]: Detect communities based on tensor factorization. This method assumes that nodes of different types have the same number of communities, and requires this number as an input. We set this number to four, the number of red communities.



(a)

(b)

(c)

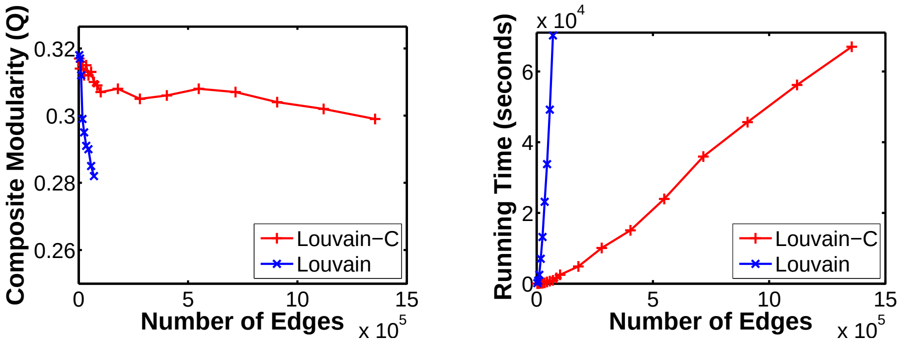
Figure 7: The NMI values achieved by different methods in the synthetic networks for (a) the red node set, (b) the green node set, and (c) the blue node set.

The results are shown in Fig. 7. We can find that Louvain-C algorithm outperforms the others by a large margin. It successfully detected the planted communities in the red, green, and blue node sets when the noise is 0%. When the noise goes up to 200%, the NMI values are still higher than 0.75. As for other methods, none of them can detect the planted communities with 100% accuracy even when the noise is 0%. In specific, the inferiority of Trans-JD and Trans-CN is due to the fact that we cannot rely on local measures (common neighbor index and Jaccard Index) to accurately calculating the similarity between nodes. Moreover, as the noise increases, the transferred unipartite networks of the two methods become so dense that almost all pairs of nodes are connected. Thus, it makes it increasingly difficult to detect communities. As a result, the performances of Trans-JD and Trans-CN plummet dramatically as the noise increases. NaiveSimp has good performance in the blue node set, but does not work well in the red and green node sets. This is because the information contained in the simplified single-relational network is sometimes incomplete. For example, in the unipartite subnetwork of green nodes there are dense edges both within and between the community G1 and G2. Thus, NaiveSimp failed to separate them and took them as a single community. The performance of MetaFac is also not so remarkable, especially in the blue node set. The reason is that this method assumes that nodes of different types have the same number of communities. However, red, green, and blue nodes have different number of planted communities. In summary, this experiment shows that our method is better than the state-of-the-art techniques in detecting the planted communities.

5.2 Scalability

In this section, we test the scalability of our method. We gradually increase the size of the synthetic network and compare Louvain-C and Louvain algorithm. Since these two algorithms are for optimizing the composite modularity, we are interested in comparing their performance in terms of running time and the composite modularity value.

Fig. 8(a) shows the composite modularity values achieved by the two algorithms. We can see that when the network size is small, Louvain algorithm is competitive with Louvain-C algorithm, but as the network size increases, Louvain-C algorithm achieved higher values than Louvain algorithm. The reason is as follows. As the network size increases, the search space increases dramatically. As a result, Louvain algorithm converged to local maxima. However, the search space can be greatly reduced, given



(a) (b)

Figure 8: (a) The composite modularity values achieved by Louvain-C and Louvain algorithm. (b) The running time of Louvain-C and Louvain algorithm. The algorithms are built in Python 3.2, and run on a PC equipped with an Intel Core i7-2600 CPU at 3.40 GHz and 32GB physical memory.

the must-be-assigned-together constraints. Thus, Louvain-C algorithm produced better results.

Fig. 8(b) shows that Louvain-C algorithm is much faster than Louvain algorithm. For example, when the network size is 70,000 edges, Louvain algorithm took 70,199.6 seconds while Louvain-C algorithm took only 911.1 seconds, a reduction of around 77 times. Thus, Louvain-C algorithm can search better solutions than Louvain algorithm in much less time.

5.3 Application in Digg Network

Finally, we apply Louvain-C algorithm to a real-world Digg network. Digg is a social news website. Digg users can submit web content as a story and other users can vote the story by “digging” it. In addition, users can add other users as their friends. We collected a subset of the stories submitted during Oct 8-15, 2010. Then we constructed a heterogeneous multi-relational network, which contains 1,191 users, 3,610 stories, 7,318 edges representing the friendship between users, and 20,941 edges representing the digging relationship between users and stories.

Louvain-C algorithm detected 90 user communities and 88 story communities in the Digg network. The composite modularity is 0.4203. We can check the reasonability of our results by looking at the topic associated with each story. In total there are 10 predesignated topics (business, entertainment, gaming, lifestyle, offbeat, politics, science, sports, technology,

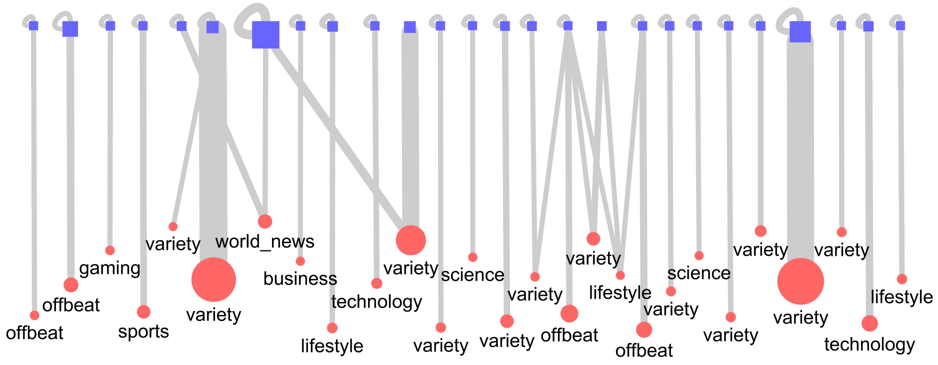


Figure 9: Visualization of the Digg network at the community level (small communities of less than 10 nodes, and sparse connections with density less than 0.05 are omitted). The story and user community are colored in red and blue, respectively. The label of a story community denotes its dominant topic (*variety* means that there is no dominant topic).

and world news), and Digg users can assign one of the topics to a story during submission. We found that many story communities have dominant topics. This means that a great many users are interested in stories with determined topics. This makes sense because people always focus on limited stories based on their personal interests.

Fig. 9 visualizes the Digg network at the community level. From this figure, we can gain knowledge about which user community is interested in which story community. We find that most user communities have only one corresponding story community, while only a few have several corresponding story communities. In addition, we can predict the interests of a user community based on the topic of its corresponding story community. This is a significant advantage of our method over NaiveSimp, Trans-CN, and Trans-JD, since they only consider one type of nodes at a time and cannot gain such information that are hidden between different types of nodes.

To better understand how one can use our method for analysis, we focus on a user community and its corresponding story community, as shown in Fig. 10. The user community contains 5 members who are densely connected to each other. The story community contains 7 members. Connections between the user community and the story community are dense. Obviously, the members of the user/story community have the similar link patterns. Thus, both communities are meaningful. Moreover, 6 members of the story community have the same topic *offbeat*. Based on this, we can predict that

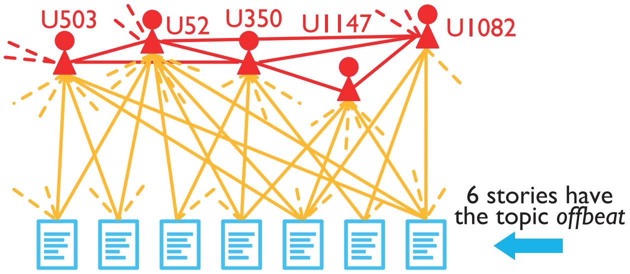


Figure 10: A user community and the corresponding story community.

these 5 users are especially interested in offbeat events. Mining such information is useful in targeted advertising.

Note that in Fig. 9 there are two user communities which have only sparse connections in between them. The identification of them is based on their link patterns with the story nodes. Clearly, the two communities cannot be detected by NaiveSimp, Trans-CN, or Trans-JD. In summary, this experiment shows that our method detected reasonable communities that agree with the natural human intuition and perception.

6 Conclusion

Previous community detection methods overwhelmingly focused on the homogeneous single-relational network which contain only one type of nodes and edges. However, Many real-world systems are naturally described as heterogeneous multi-relational networks which contain multiple types of nodes and edges. In this paper, we proposed the first modularity measure — the composite modularity for evaluating partitions of a heterogeneous multi-relational network into communities. The key idea of our composite modularity is to decompose a heterogeneous multi-relational network into multiple subnetworks, and integrate the modularities in each subnetworks. Compared to modularity and its variants for single-relational networks, our composite modularity can effectively utilize multi-faceted information in a heterogeneous multi-relational network and comprehensively evaluate a partition. We developed a fast algorithm for optimizing the composite modularity and detecting the communities in heterogeneous multi-relational networks. In short, our composite modularity optimization method has the following advantages:

- It is consistent with the modularity optimization method for detecting communities in homogeneous single-relational networks;

- It is applicable to networks of general structure, which may even contain hyper-edges;
- It is parameter free and can automatically detect communities without any a priori knowledge such as the number of communities;
- It is fast and scalable to large-scale networks.

A notable drawback of modularity optimization is the resolution limit, which refers to the incapability of detecting small communities in large-scale networks [14, 17, 23]. Although it has not been shown, we expect that the composite modularity optimization method introduced in this work has a similar resolution limit. The resolution limit is attributed to the globalization of the null model, which assumes that any node can connect to any others. Thus, redefining the null model and restricting such globalization are a possible way to solve the resolution limit. This is left for our future work.

Acknowledgements

We gratefully thank Mr. Pen-Lin Chang for collecting the Digg data.

References

[1] Arenas, A., Fernández, A., and Gómez, S., Analysis of the structure of complex networks at different resolution levels, *New Journal of Physics* **10** (2008) 053039.

[2] Barber, M. J., Modularity and community detection in bipartite network, *Phys. Rev. E* **76** (2007) 066102.

[3] Barber, M. J. and Clark, J. W., Detecting network communities by propagating labels under constraints, *Phys. Rev. E* **80** (2009) 026129.

[4] Blondel, V. D., Guillaume, J. L., Lambiotte, R., and Lefebvre, E., Fast unfolding of communities in large networks, *J. Stat. Mech.* (2008) P10008.

[5] Bogdanov, P., Larusso, N. D., and Singh, A., Towards community discovery in signed collaborative interaction networks, in *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops* (Sydney, Australia, 2010), pp. 288–295.

[6] Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikolski, Z., and Wagner, D., On modularity — np-completeness and beyond, Technical Report 2006-19, ITI Wagner, Faculty of Informatics, Universität Karlsruhe (2006).

[7] Cho, H., Dhillon, I. S., Guan, Y., and Sra, S., Minimum sum-squared residue co-clustering of gene expression data, in *Proceedings of the 4th SIAM International Conference on Data Mining* (Lake Buena Vista, FL, USA, 2004), pp. 114–125.

[8] Clauset, A., Newman, M. E. J., and Moore, C., Finding community structure in very large networks, *Phys. Rev. E* **70** (2004) 066111.

[9] Comar, P. M., Tan, P. N., and Jain, A. K., A framework for joint community detection across multiple related networks, *Neurocomputing* **76** (2012) 93–104.

[10] Danon, L., Duch, J., D.-Guilera, A., and Arenas, A., Comparing community structure identification, *J. Stat. Mech.* (2005) P09008.

[11] Dhillon, I. S., Mallela, S., and Modha, D. S., Information-theoretic co-clustering, in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC, USA, 2003), pp. 89–98.

[12] Duch, J. and Arenas, A., Community detection in complex networks using extremal optimization, *Phys. Rev. E* **72** (2005) 027104.

[13] Fortunato, S., Community detection in graphs, *Physics Reports* **486** (2010) 75–174.

[14] Fortunato, S. and Barthélemy, M., Resolution limit in community detection, *Proc. Natl. Acad. Sci. USA* **104** (2007) 36–41.

[15] Fred, A. L. N. and Jain, A. K., Robust data clustering, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Madison, WI, USA, 2003), pp. 128–133.

[16] Girvan, M. and Newman, M. E. J., Community structure in social and biological networks, *Proc. Natl. Acad. Sci. USA* **99** (2002) 7821–7826.

[17] Good, B. H., de Montjoye, Y.-A., and Clauset, A., The performance of modularity maximization in practical contexts, *Phys. Rev. E* **81** (2010) 046106.

[18] Guimerà, R., S.-Pardo, M., and Amaral, L. A. N., Module identification in bipartite and directed networks, *Phys. Rev. E* **76** (2007) 036102.

[19] He, D., Liu, J., Yang, B., Huang, Y., Liu, D., and Jin, D., An ant-based algorithm with local optimization for community detection in large-scale networks, *Advances in Complex Systems* **15** (2012) 1250036.

[20] Kernighan, B. and Lin, S., An efficient heuristic procedure to partition graphs, *Bell Syst. Tech. J.* **49** (1970) 291–307.

[21] Lancichinetti, A. and Fortunato, S., Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, *Phys. Rev. E* **80** (2009) 016118.

[22] Lancichinetti, A. and Fortunato, S., Community detection algorithms: a comparative analysis, *Phys. Rev. E* **80** (2009) 056117.

[23] Lancichinetti, A. and Fortunato, S., Limits of modularity maximization in community detection, *Phys. Rev. E* **84** (2011) 066122.

[24] Lancichinetti, A., Fortunato, S., and Radicchi, F., Benchmark graphs for testing community detection algorithms, *Phys. Rev. E* **78** (2008) 046110.

[25] Lin, Y. R., Sun, J., Castro, P., Konuru, R., Sundaram, H., and Kelliher, A., Metafac: community discovery via relational hypergraph factorization, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France, 2009), pp. 527–535.

[26] Liu, X. and Murata, T., Advanced modularity-specialized label propagation algorithm for detecting communities in networks, *Physica A* **389** (2010) 1493–1500.

[27] Liu, X. and Murata, T., Detecting communities in k-partite k-uniform (hyper)networks, *Journal of Computer Science and Technology* **26** (2011) 778–791.

[28] Long, B., Wu, X., Zhang, Z., and Yu, P. S., Community learning by graph approximation, in *Proceedings of the 7th IEEE International Conference on Data Mining* (Cambridge, MA, USA, 2007), pp. 232–241.

[29] Long, B., Zhang, Z., Yu, P. S., and Xu, T., Clustering on complex graphs, in *Proceedings of the 23rd National Conference on Artificial Intelligence* (Chicago, IL, USA, 2008), pp. 659–664.

[30] Long, B., Zhang, Z. M., and Yu, P. S., Co-clustering by block value decomposition, in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, IL, USA, 2005), pp. 635–640.

[31] Medus, A., Acuna, G., and Dorso, C. O., Detection of community structures in networks via global optimization, *Physica A* **358** (2005) 593–604.

[32] Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J. P., Community structure in time-dependent, multiscale, and multiplex networks, *Science* **328** (2010) 876–878.

[33] Murata, T., Detecting communities from tripartite networks, in *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, NC, USA, 2010), pp. 1159–1160.

[34] Murata, T. and Ikeya, T., A new modularity for detecting one-to-many correspondence of communities in bipartite networks, *Advances in Complex Systems* **13** (2010) 19–31.

[35] Neubauer, N. and Obermayer, K., Towards community detection in k-partite k-uniform hypergraphs, in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems, Workshop on Analyzing Networks and Learning with Graphs* (Whistler, BC, Canada, 2009).

[36] Newman, M. E. J., Fast algorithm for detecting community structure in networks, *Phys. Rev. E* **69** (2004) 066133.

[37] Newman, M. E. J., Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E* **74** (2006) 036104.

[38] Newman, M. E. J., Communities, modules and large-scale structure in networks, *Nature Physics* **8** (2011) 25–31.

[39] Newman, M. E. J. and Girvan, M., Finding and evaluating community structure in networks, *Phys. Rev. E* **69** (2004) 026113.

[40] Orman, G. K., Labatut, V., and Cherifi, H., Comparative evaluation of community detection algorithms: a topological approach, *J. Stat. Mech.* (2012) P08001.

[41] Popescul, A., Flake, G. W., Lawrence, S., Ungar, L. H., and Giles, C. L., Clustering and identifying temporal trends in document databases, in *Proceedings of the IEEE Advances in Digital Libraries* (Bethesda, MD, USA, 2000), pp. 173–182.

[42] Reichardt, J. and Bornholdt, S., Statistical mechanics of community detection, *Phys. Rev. E* **74** (2006) 016110.

[43] Schuetz, P. and Caflisch, A., Efficient modularity optimization by multistep greedy algorithm and vertex refinement, *Phys. Rev. E* **77** (2008) 046112.

[44] Schuetz, P. and Caflisch, A., Multistep greedy algorithm identifies community structure in real-world and computer-generated networks, *Phys. Rev. E* **78** (2008) 026112.

[45] Scott, J., *Social Network Analysis: A Handbook*, 2nd edn. (Sage Publications, Newberry Park, CA, 2000).

[46] Shi, C., Yan, Z., Wang, Y., Cai, Y., and Wu, B., A genetic algorithm for detecting communities in large-scale complex network, *Advances in Complex Systems* **13** (2010) 3–17.

[47] Sun, J., Faloutsos, C., Papadimitriou, S., and Yu, P. S., Graphscope: parameter-free mining of large time-evolving graphs, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Jose, CA, USA, 2007), pp. 687–696.

[48] Sun, Y., Aggarwal, C. C., and Han, J., Relation strength-aware clustering of heterogeneous information networks with incomplete attributes, *Proceedings of the VLDB Endowment* **5** (2012) 394–405.

[49] Sun, Y., Yu, Y., and Han, J., Ranking-based clustering of heterogeneous information networks with star network schema, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France, 2009), pp. 797–806.

[50] Tang, L., Liu, H., and Zhang, J., Identifying evolving groups in dynamic multimode networks, *IEEE Transactions on Knowledge and Data Engineering* **24** (2012) 72–85.

[51] Tang, L., Wang, X., and Liu, H., Community detection via heterogeneous interaction analysis, *Data Mining and Knowledge Discovery* **25** (2012) 1–33.

[52] Traag, V. A. and Bruggeman, J., Community detection in networks with positive and negative links, *Phys. Rev. E* **80** (2009) 036115.

[53] Wakita, K. and Tsurumi, T., Finding community structure in mega-scale social networks, in *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada, 2007), pp. 1275–1276.

[54] Yang, B., Cheung, W. K., and Liu, J., Community mining from signed social networks, *IEEE Transactions on Knowledge and Data Engineering* **19** (2007) 1333–1348.