New Generation Computing, 34(2016)153-176
Ohmsha, Ltd. and Springer

Community Detection in Multi-Partite Multi-Relational Networks Based on Information Compression

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. JST CREST
K's Gobancho, 7, Gobancho, Chiyoda, Tokyo, 102-0076 JAPAN
3. Wuhan University of Technology
122 Luoshi Road, Wuhan, Hubei, 430070 CHINA
{tsinllew@ai.cs, liu.w@ai.cs, murata@cs, wakita@is}.titech.ac.jp

Received July 19 2014
Revised manuscript received April 21 2015

Abstract Community detection in uni-partite single-relational networks which contain only one type of nodes and edges has been extensively studied in the past decade. However, many real-world systems are naturally described as multi-partite multi-relational networks which contain multiple types of nodes and edges. In this paper, we propose an information compression based method for detecting communities in such networks. Specifically, based on the minimum description length (MDL) principle, we propose a quality function for evaluating partitions of a multi-partite multi-relational network into communities, and develop a heuristic algorithm for optimizing the quality function. We demonstrate that our method outperforms the state-of-the-art techniques in both synthetic and real-world networks.

Keywords: Community Detection, Graph Mining, Clustering, Multi-Partite Multi-Relational Network, Information Compression.

§1 Introduction

Many social, biological, and information systems can be naturally represented as networks—The elementary units of the system are reduced to nodes, while their relations and interactions are pictured as edges (or called links). In recent years, there has been a surge of interests in analysis of networked

datasets. Researchers found that the structure of most networks, beneath the intrinsic disorder due to the stochastic character of their generation mechanisms, reveals a high degree of organization. In particular, nodes with similar properties or functions have a higher chance to be linked to each other and tend to form highly cohesive subnetworks, which are called *communities* (also *modules* or *clusters*). Examples of communities are groups of people with common interests in social networks,[1] collections of Web pages on closely related topics,[2,3] biochemical pathways in metabolic networks,[4,5] and etc. Detecting communities in networks may help to identify functional subunits of the system and provide insight into how the system is internally organized.[6,7] The community structure of a network can also be a powerful visual representation of the system—Instead of visualizing all the nodes and edges of the network (which is impossible on large systems), one could display its communities and their mutual interactions, obtaining a far more compact and refined description.[8,9]

Previous studies on community detection mainly focus on uni-partite single-relational networks which contain only one type of nodes and edges. Many real-world systems, however, are described as *multi-partite* (multiple types of nodes) *multi-relational* (multiple types of edges) networks. Take a 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 multi-partite multi-relational network, which contains three types of nodes (users, photos, and tags) and three types of edges representing the above relations.

Traditionally we rely on only one type of edges, and simplify a multi-partite multi-relational network to a single-relational network. However, this method might be insufficient to determine community membership accurately. Take the above Flickr network as an example. The data about users might be incomplete and noisy, and some active users have thousands of friends while

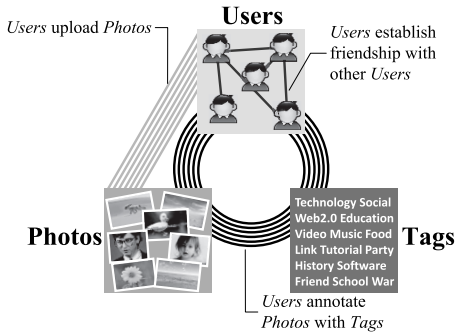


Fig. 1 Describing Flickr as a multi-partite multi-relational network. This network contains three types of nodes: 1) users, 2) photos, 3) tags; and three types of edges: 1) the edges representing friendship between users, 2) the edges representing that users unload photos, 3) the 3-way hyper-edges representing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

some others have no friends at all. Consequently, we cannot obtain real user communities based on the friendship alone. In this scenario, effectively utilizing various types of edges would enhance the community detection results.

To detect communities in a multi-partite multi-relational network, researchers have proposed approaches based on ranking[10] and on non-negative matrix factorization.[11] However, they are restricted to specific subclass of networks. In addition, Lin et al. proposed a method based on tensor factorization.[12] A drawback is that this method require a priori knowledge about the number of communities, limiting its usage in inferring the latent organization of a real system. Recently, Liu et al. proposed a composite modularity maximization method.[13] The idea is to decompose a multi-partite multi-relational network into multiple single-relational subnetworks, each composed of one type of edge and the incident nodes. Then, a composite modularity which is an integration of the modularity in each subnetwork is proposed for evaluating community structure in the multi-partite multi-relational network. However, due to the deficiency in definition, the composite modularity cannot handle communities of many-to-many correspondence (i.e., a community of one node type x_1 can have dense edges to many communities of another node type x_2, and conversely, a community of type x_2 can have dense edges to many communities of type x_1).

In this paper, we propose an information compression based method for community detection in multi-partite multi-relational networks. The idea is to convert the community detection problem to a problem of finding an efficient compression of the network's structure. More precisely, based on the minimum description length (MDL) principle[14] which accounts for the best compression of network structure data, we propose a quality function for evaluating partitions of a multi-partite multi-relational network into communities, and develop a heuristic algorithm for optimizing the quality function. Our new method overcomes the limitations of existing methods and has the following advantages:

- General: it is able to handle broad families of multi-partite multi-relational networks and is applicable to communities of many-to-many correspondence.
- Parameter-free: it can automatically detect communities, without any a priori knowledge like the numbers of communities.
- Accurate: it is more sensitive than state-of-the-art techniques.

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 multi-partite multi-relational network. Section 4 introduces our new method. Section 5 presents experimental results, followed by a conclusion in Section 6.

§2 Related Work

The problem of community detection in single-relational networks has been extensively studied in the last decade.[6,15–18] Researchers have proposed various methods, such as modularity maximization,[19–22] spectral methods,[23,24] and methods based on statistical inference.[25,26] In addition, there are studies

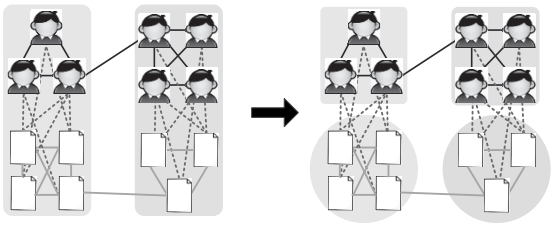


Fig. 2 In the left panel, the network is partitioned into two communities, each contain both authors and papers. If we take authors and papers separately, this partition implies that authors and papers have the same number of communities, as shown in the right panel.

on community detection in uni-partite multi-relational networks. For example, Mucha et al. proposed a multiplex model for describing a uni-partite multi-relational network and developed a method based on maximizing a generalized modularity.[27] Tang et al. proposed methods based on matrix approximation[28] and spectral analysis.[29]

In recent years, some researchers addressed the problem of community detection in multi-partite multi-relational networks. Comar et al. developed a method based on non-negative matrix factorization.[11] However, their work is restricted to a specific subclass of networks which contain 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 as the star network schema.[10] Thus both of these two methods are not applicable to general networks with any possible structure. As for the problem in a general multi-partite 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.[30] However, when the structure of a network becomes complex, we cannot find a 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.[12] They suppose that a community contain nodes of different types. If we take nodes of different types separately, this implies that nodes of different types have the same number of communities, as illustrated in Fig. 2. Unfortunately, this situation is rarely seen in real-world scenario. Another drawback of Lin's method is that it requires a priori knowledge about the number of communities, limiting its usage in inferring the latent organization of a real system. In another important work,[13] the authors decomposed a multi-partite multi-relational network into multiple single-relational subnetworks, and proposed the composite modularity as an integration of the modularity in each subnetwork. However, due to the deficiency in definition, the composite modularity cannot handle communities of many-to-many correspondence.

The idea of using information compression can be traced back to the information-theoretic co-clustering algorithm.[31] Researchers have used informa-

tion compression for community detection. For example, Rosvall and Bergstrom proposed a method for community detection in uni-partite single-relational networks.[32] Sun et al.[33] and Liu et al.[34] developed methods for community detection in multi-partite single-relational networks (bi-partite network, k-partite network) respectively. In this paper, we followed Rosvall and Bergstrom's idea of converting the community detection problem to a problem of finding an efficient lossy compression of the network's structure based on the MDL principle. We generalized their method for community detection in uni-partite single-relational networks to this problem in multi-partite multi-relational networks.

§3 Problem Formulation

In this section, we formulate the problem of community detection in a multi-partite multi-relational network. For clearness, we list major notations in Table 1. Now suppose a multi-partite 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:

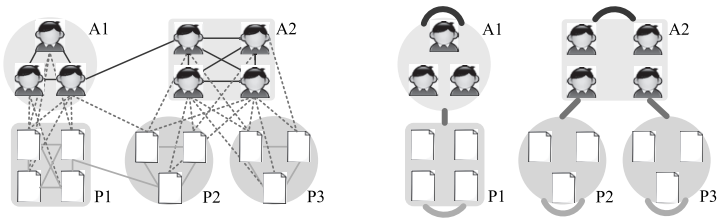
1. There exists a $x \in \{1, 2, \cdots, r\}$, such that $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x]} \times \mathbf{V}^{[x]}$, i.e., $\mathbf{E}^{[y]}$ is a set of edges that link to nodes of the same type.
2. There exists $x_1, x_2, \cdots, x_k \in \{1, 2, \cdots, r\}$ $(k \leq r)$ which are not equal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

We are interested in link pattern based communities in $\mathbf{G}$.[35,36] A link-pattern based community is a group of nodes which have similar link patterns. In other words, the nodes within a community link to other nodes in similar ways. Figure 3(a) shows a multi-partite 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

Table 1 Notations for a Multi-partite 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
$\mathbf{A}^{[y]}$	The connectivity array of $\mathbf{E}^{[y]}$
$n^{[x]}$	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 $v_i^{[x]}$
$\mathbf{V}_\alpha^{[x]}$	The α-th community in $\mathbf{V}^{[x]}$

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



(a) (b)

Fig. 3 (a) The link pattern based community.
(b) The link patterns of the communities.

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 link to nodes in A1 and P1, and sparsely link to nodes in A2, P2, and P3. Similar interpretation applies to other communities. Figure 3(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 a real community.

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 Information Compression Based Method

In this section, we propose the information compression based method for the problem formulated in Section 3. When we describe a network at the community level, we are highlighting certain regularities of the network structure while filtering out relatively unimportant details. Thus, a description of a network at the community level can be viewed as a lossy compression of the network's structure.[32] Based on this idea, we convert the problem of community detection to a problem of finding an efficient compression of the network's structure. Specifically, based on the MDL principle[14] which accounts for the best compression of network structure data, we propose a quality function for evaluating partitions of a multi-partite multi-relational network $\mathbf{G}$ into communities. Then, we develop a heuristic algorithm for optimizing the quality function and finding the best possible partition.

4.1 Quality Function

In this subsection, we show how to compress the structure information of a multi-partite multi-relational network $\mathbf{G}$, in order to formulate our quality function. Figure 4 shows a communication process of transmitting structure information of $\mathbf{G}$. Suppose a signaler knows the structure information $\mathbf{X}$ which describes the links of $\mathbf{G}$ at the node level, and he aims to transmit much of $\mathbf{X}$ to a receiver. To do so, the signaler makes a partition of $\mathbf{G}$ into communities

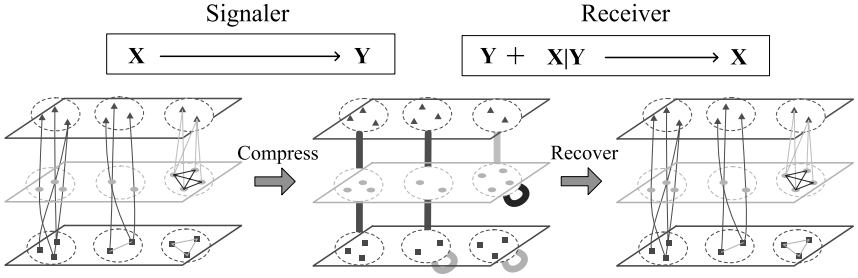


Fig. 4 Communication Between a Signaler and a Receiver.

and encodes $\mathbf{X}$ as compressed information $\mathbf{Y}$ which describes the links at the community level.

The structure information $\mathbf{X}$ can be represented by s arrays, i.e., $\mathbf{X} = \{\mathbf{A}^{[1]} \cup \mathbf{A}^{[2]} \cup \cdots \cup \mathbf{A}^{[s]}\}$, where $\mathbf{A}^{[y]}$ describes the link structure of $\mathbf{E}^{[y]}$ at the node level ($y \in \{1, 2, \cdots, s\}$), and satisfies the following conditions:

Case 1 : $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x]} \times \mathbf{V}^{[x]}$ ($x \in \{1, 2, \cdots, r\}$). Suppose $n^{[x]} = |\mathbf{V}^{[x]}|$ is the number of nodes in $\mathbf{V}^{[x]}$. Then, $\mathbf{A}^{[y]}$ is a $n^{[x]} \times n^{[x]}$ array, with elements

$$A_{ij}^{[y]} = \begin{cases} 1 & \text{if } (v_i^{[x]}, v_j^{[x]}) \in \mathbf{E}^{[y]}; \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $v_i^{[x]}$ denotes the i-th node in $\mathbf{V}^{[x]}$.

Case 2 : $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x_1]} \times \mathbf{V}^{[x_2]} \times \cdots \times \mathbf{V}^{[x_k]}$ ($x_1, x_2, \cdots, x_k \in \{1, 2, \cdots, r\}$ and $k \leq r$). Then, $\mathbf{A}^{[y]}$ is a $n^{[x_1]} \times n^{[x_2]} \times \cdots \times n^{[x_k]}$ array, with elements

$$A_{i_1 i_2 \cdots i_k}^{[y]} = \begin{cases} 1 & \text{if } (v_{i_1}^{[x_1]}, v_{i_2}^{[x_2]}, \cdots, v_{i_k}^{[x_k]}) \in \mathbf{E}^{[y]}; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The compressed information is $\mathbf{Y} = \{\mathcal{L}^{[1]} \cup \mathcal{L}^{[2]} \cup \cdots \cup \mathcal{L}^{[r]} \cup \mathbf{M}^{[1]} \cup \mathbf{M}^{[2]} \cup \cdots \cup \mathbf{M}^{[s]}\}$. Here $\mathcal{L}^{[x]} = \{l_1^{[x]}, l_2^{[x]}, \cdots, l_{n^{[x]}}^{[x]}\}$ is a community membership vector for $\mathbf{V}^{[x]}$ ($x \in \{1, 2, \cdots, r\}$), where $l_i^{[x]}$ indicates the community of $v_i^{[x]}$. $\mathbf{M}^{[y]}$ describes the link structure of $\mathbf{E}^{[y]}$ at the community level ($y \in \{1, 2, \cdots, s\}$), and satisfies the following conditions:

Case 1 : $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x]} \times \mathbf{V}^{[x]}$ ($x \in \{1, 2, \cdots, r\}$). Suppose $c^{[x]}$ is the number of communities in $\mathbf{V}^{[x]}$. Then, $\mathbf{M}^{[y]}$ is a $c^{[x]} \times c^{[x]}$ array, with elements

$$M_{\alpha\beta}^{[y]} = \sum_{v_i^{[x]} \in \mathbf{V}_\alpha^{[x]}} \sum_{v_j^{[x]} \in \mathbf{V}_\beta^{[x]}} A_{ij}^{[y]}, \tag{3}$$

where $\mathbf{V}_\alpha^{[x]} = \{v_i^{[x]} | l_i^{[x]} = \alpha\}$ denotes the α-th community in $\mathbf{V}^{[x]}$.

Case 2 : $\mathbf{E}^{[y]} \subseteq \mathbf{V}^{[x_1]} \times \mathbf{V}^{[x_2]} \times \cdots \times \mathbf{V}^{[x_k]}$ ($x_1, x_2, \cdots, x_k \in \{1, 2, \cdots, r\}$ and

$k \leq r$). Then, $\mathbf{M}^{[y]}$ is a $c^{[x_1]} \times c^{[x_2]} \times \cdots \times c^{[x_k]}$ array, with elements

$$M^{[y]}_{\alpha_1\alpha_2\cdots\alpha_k} = \sum_{v^{[x_1]}_{i_1} \in \mathbf{V}^{[x_1]}_{\alpha_1}} \sum_{v^{[x_2]}_{i_2} \in \mathbf{V}^{[x_2]}_{\alpha_2}} \cdots \sum_{v^{[x_k]}_{i_k} \in \mathbf{V}^{[x_k]}_{\alpha_k}} A^{[y]}_{i_1 i_2 \cdots i_k} \tag{4}$$

The description length of $\mathbf{Y}$ (in bits) can be measured in the following way. As for $\mathcal{L}^{[x]}$, there are $n^{[x]}$ elements and each element $l^{[x]}_i \in \{0, 1, \cdots, c^{[x]}\}$. Thus, the description length of $\mathcal{L}^{[x]}$ is

$$\mathrm{Len}(\mathcal{L}^{[x]}) = n^{[x]} \log c^{[x]}, \tag{5}$$

where the logarithm is taken in base 2.[*2] As for $\mathbf{M}^{[y]}$, there are two cases:

Case 1 : $\mathbf{M}^{[y]}$ is a $c^{[x]} \times c^{[x]}$ array for describing link structure of communities in $\mathbf{V}^{[x]}$. Note that $\mathbf{M}^{[y]}$ is symmetric and the number of independent elements is $c^{[x]}(c^{[x]}+1)/2$. Each element $M^{[y]}_{\alpha\beta} \in 0, 1, \cdots, m^{[y]}$, where $m^{[y]} = |\mathbf{E}^{[y]}|$ is the number of edges in $\mathbf{E}^{[y]}$. Thus, the description length of $\mathbf{M}^{[y]}$ is

$$\mathrm{Len}(\mathcal{M}^{[y]}) = \frac{c^{[x]}(c^{[x]}+1)}{2} \log(m^{[y]}+1). \tag{6}$$

Case 2 : $\mathbf{M}^{[y]}$ is a $c^{[x_1]} \times c^{[x_2]} \times \cdots \times c^{[x_k]}$ array for describing link structure of communities in $\mathbf{V}^{[x_1]}, \mathbf{V}^{[x_2]}, \cdots, \mathbf{V}^{[x_k]}$. Note that each element $M^{[y]}_{\alpha_1\alpha_2\cdots\alpha_k} \in \{0, 1, \cdots, m^{[y]}\}$. Thus, the description length of $\mathbf{M}^{[y]}$ is

$$\mathrm{Len}(\mathcal{M}^{[y]}) = c^{[x_1]} c^{[x_2]} \cdots c^{[x_k]} \log(m^{[y]}+1). \tag{7}$$

As a result, the description length of $\mathbf{Y}$ is

$$\mathrm{Len}(\mathbf{Y}) = \sum_{x=1}^{r} \mathrm{Len}(\mathcal{L}^{[x]}) + \sum_{y=1}^{s} \mathrm{Len}(\mathcal{M}^{[y]}). \tag{8}$$

After receiving $\mathbf{Y}$, the receiver can estimate $\mathbf{X}$ by creating a number of candidates. For example, suppose the receiver is to estimate $\mathbf{A}^{[y]}$.

Case 1 : $\mathbf{M}^{[y]}$ is a $c^{[x]} \times c^{[x]}$ array for describing link structure of communities in $\mathbf{V}^{[x]}$. With $\mathcal{L}^{[x]}$ and $\mathbf{M}^{[y]}$, the receiver knows: 1) the number of edges $M^{[y]}_{\alpha\beta}$ between communities $\mathbf{V}^{[x]}_{\alpha}$ and $\mathbf{V}^{[x]}_{\beta}$; 2) which nodes are in communities $\mathbf{V}^{[x]}_{\alpha}$ and $\mathbf{V}^{[x]}_{\beta}$. What he does not know is which nodes are incident to these $M^{[y]}_{\alpha\beta}$ edges. The number of possibilities of recovering

[*2] The description length of encoding an unknown positive integer z is $\log_2 z + \log_2\log_2 z + \cdots$, where only the positive terms are retained.[37] Here we just make a rough calculation.

these edges can be given by

$$\begin{cases} \binom{n_\alpha^{[x]}(n_\alpha^{[x]}-1)/2}{M_{\alpha\alpha}^{[y]}} & \text{if } \alpha=\beta \text{ (suppose no self-loop is allowed);} \\ \binom{n_\alpha^{[x]}n_\beta^{[x]}}{M_{\alpha\beta}^{[y]}} & \text{if } \alpha\neq\beta, \end{cases} \tag{9}$$

where the parentheses denote the binomial coefficient, and $n_\alpha^{[x]}=|\mathbf{V}_\alpha^{[x]}|$ is the number of nodes in $\mathbf{V}_\alpha^{[x]}$. Thus, the number of candidates for estimating $\mathbf{A}^{[y]}$ is

$$\mathrm{NumCdt}(\mathbf{A}^{[y]})=\prod_{\alpha=1}^{c^{[x]}}\binom{n_\alpha^{[x]}(n_\alpha^{[x]}-1)/2}{M_{\alpha\alpha}^{[y]}}\prod_{\alpha>\beta}\binom{n_\alpha^{[x]}n_\beta^{[x]}}{M_{\alpha\beta}^{[y]}} \tag{10}$$

Case 2 : $\mathbf{M}^{[y]}$ is a $c^{[x_1]}\times c^{[x_2]}\times\cdots\times c^{[x_k]}$ array for describing link structure of communities in $\mathbf{V}^{[x_1]},\mathbf{V}^{[x_2]},\cdots,\mathbf{V}^{[x_k]}$. With $\mathcal{L}^{[x_1]},\mathcal{L}^{[x_2]},\cdots,\mathcal{L}^{[x_k]}$ and $\mathbf{M}^{[y]}$, the receiver knows: 1) the number of edges $M_{\alpha_1\alpha_2\cdots\alpha_k}^{[y]}$ between communities $\mathbf{V}_{\alpha_1}^{[x_1]},\mathbf{V}_{\alpha_2}^{[x_2]},\cdots,\mathbf{V}_{\alpha_k}^{[x_k]}$; 2) which nodes are in communities $\mathbf{V}_{\alpha_1}^{[x_1]},\mathbf{V}_{\alpha_2}^{[x_2]},\cdots,\mathbf{V}_{\alpha_k}^{[x_k]}$. What he does not know is which nodes are incident to these $M_{\alpha_1\alpha_2\cdots\alpha_k}^{[y]}$ edges. The number of possibilities of recovering these edges can be given by

$$\binom{n_{\alpha_1}^{[x_1]}n_{\alpha_2}^{[x_2]}\cdots n_{\alpha_k}^{[x_k]}}{M_{\alpha_1\alpha_2\cdots\alpha_k}^{[y]}}. \tag{11}$$

Thus, the number of candidates for estimating $\mathbf{A}^{[y]}$ is

$$\mathrm{NumCdt}(\mathbf{A}^{[y]})=\prod_{\alpha_1=1}^{c^{[x_1]}}\prod_{\alpha_2=1}^{c^{[x_2]}}\cdots\prod_{\alpha_k=1}^{c^{[x_k]}}\binom{n_{\alpha_1}^{[x_1]}n_{\alpha_2}^{[x_2]}\cdots n_{\alpha_k}^{[x_k]}}{M_{\alpha_1\alpha_2\cdots\alpha_k}^{[y]}}. \tag{12}$$

As a result, the total number of candidates for estimating $\mathbf{X}$ is

$$\prod_{y=1}^{s}\mathrm{NumCdt}(\mathbf{A}^{[y]}). \tag{13}$$

To exactly recover $\mathbf{X}$, the receiver needs additional information about which of the candidates is the right one. The description length of the additional information is

$$\mathrm{Len}(\mathbf{X}|\mathbf{Y})=\sum_{y=1}^{s}\log[\mathrm{NumCdt}(\mathbf{A}^{[y]})]. \tag{14}$$

The objective is for the signaler to transmit the least information while the receiver gains the most (i.e., the receiver needs the least additional information to recover $\mathbf{X}$). This is apparently a dilemma. For example, if the signaler makes a partition of $\mathbf{V}^{[x]}$ into $n^{[x]}$ communities ($x\in\{1,2,\cdots,r\}$), meaning one community for each node, there would be no compression on $\mathbf{Y}$. Thus, the

receiver can recover $\mathbf{X}$ exactly without any additional information, while the signaler has to transmit the most. On the other hand, if the signaler makes a partition of $\mathbf{V}^{[x]}$ into only one community ($x \in \{1, 2, \cdots, r\}$), $\mathbf{Y}$ would be the most compressed. Thus, the signaler transmits the least information, while the receiver needs much additional information to recover $\mathbf{X}$.

If the signaler makes a "good" partition of the network into link pattern based communities, he can highlight certain regularities (the similarity in terms of link pattern for nodes in the same community) and filter out relatively unimportant details. Intuitively, compression based on this partition would achieve a trade-off between $\mathrm{Len}(\mathbf{Y})$ and $\mathrm{Len}(\mathbf{X}|\mathbf{Y})$. According to the minimum description length (MDL) principle,[14, 32, 34]

$$\mathrm{Q}^{\mathrm{MDL}}(\mathcal{L}) = \mathrm{Len}(\mathbf{Y}) + \mathrm{Len}(\mathbf{X}|\mathbf{Y}) \tag{15}$$

would get the minimum value. This is our quality function for evaluating $\mathcal{L}$. It is clear that the lower the $\mathrm{Q}^{\mathrm{MDL}}$, the better the $\mathcal{L}$.

4.2 Optimization Algorithm

We develop a greedy algorithm called *InfoCom* for minimizing $\mathrm{Q}^{\mathrm{MDL}}$ and detecting communities in multi-partite multi-relational networks. InfoCom relies on two well known local search heuristics: node moving[21, 38] and community joining.[19, 20, 39, 40] The node moving heuristic repeatedly moves an individual node to a different community, which can bring a decrease in $\mathrm{Q}^{\mathrm{MDL}}$. The community joining heuristic repeatedly joins a pair of communities, which can produce a decrease in $\mathrm{Q}^{\mathrm{MDL}}$. These two heuristics are often combined to form hybrid algorithms for modularity maximization, and InfoCom is built on ideas of these existing algorithms.[21, 22]

As shown in Algorithm 1, InfoCom can be divided into two iterative phases. In this algorithm, each node is associated with a label, indicating its community ID. Initially we assign each node a unique label, implying that each singleton node constitutes a community. Then, we enter Phase 1, where we repeatedly update node labels to minimize $\mathrm{Q}^{\mathrm{MDL}}$. Suppose we are to update $v_i^{[x]}$'s label. The rule is to replace the old label by one which produces the greatest decrease of $\mathrm{Q}^{\mathrm{MDL}}$; if no new label produces a decrease of $\mathrm{Q}^{\mathrm{MDL}}$, we keep the old label unchanged. This updating process is done sequentially for each node and then repeated round by round, until no decrease of $\mathrm{Q}^{\mathrm{MDL}}$ can be attained. At the end, Phase 1 would converge to a local minimum of $\mathrm{Q}^{\mathrm{MDL}}$, and the node labels would bring a temporal community partition. Note that Phase 1 essentially equals to the node moving heuristic, since updating a node's label to a new one implies moving this node to a new community.

In Phase 2, we try to escape the local minimum of Phase 1 by the community joining heuristic. To do this efficiently, we follow the approach used in Louvain algorithm.[21] Specifically, we first build a reduced network $\bar{\mathbf{G}}$, where each node corresponds to a community in Phase 1.[41] Then, we assign each node in $\bar{\mathbf{G}}$ a unique label and update labels in the same way as Phase 1. Note that now a node corresponds to a community, and thus updating a node's label essentially

equals to joining two communities.

After Phase 2 is finished, we would come back to $\mathbf{G}$ and restart Phase 1 again. That is, we retrieve each node's label from the corresponding label in $\bar{\mathbf{G}}$, and then update labels to reach another local minimum. With these two phases repeated iteratively, a fairly good result which converges at both the local minima of node moving heuristic and community joining heuristic can finally be obtained. Also note that $\mathrm{Q}^{\mathrm{MDL}}$ keeps decreasing or unchanged during each label updating, so InfoCom is essentially a greedy algorithm.

Algorithm 1 Detecting communities in a multi-partite multi-relational network $\mathbf{G}$ by minimizing $\mathrm{Q}^{\mathrm{MDL}}$.

Input: Multi-partite multi-relational network $\mathbf{G}$
Output: Partition $\mathcal{L}$

```
begin
    Assign each node in G a unique label
    repeat
        // Phase 1
        repeat
            Sequentially update node labels in G
        until a local minimum of Q^MDL
        // Phase 2
        Build a reduced network Ḡ from G
          Assign each node in Ḡ a unique label
          repeat
            Sequentially update node labels in Ḡ
        until a local minimum of Q^MDL
        Retrieve node labels in G from the corresponding one in Ḡ
    until no change in Q^MDL
end
```

In implementation, the numbers of edges between communities, namely $\mathbf{M}^{[y]}$, are kept in real-time. Other stored data include the connectivity arrays $\mathbf{A}^{[y]}$, the community membership vectors $\mathcal{L}^{[x]}$, and the number of nodes in each community $n_\alpha^{[x]}$. The most computationally intensive step of this algorithm is to update node labels. To update $v_i^{[x]}$'s label l^{old}, we consider l^{old} and the labels of $v_i^{[x]}$'s counterparts ($v_i^{[x]}$'s 1-hop and 2-hop neighbors which are of the same node type as $v_i^{[x]}$) as candidate labels. The new label is calculated as

$$l^{\mathrm{new}} = \underset{l \in \{\text{candidate labels}\}}{\arg\min} [\mathrm{Q}^{\mathrm{MDL}}(l_i^{[x]} = l) - \mathrm{Q}^{\mathrm{MDL}}(l_i^{[x]} = l^{\mathrm{old}})]. \tag{16}$$

Note that the second term at the right hand side of Equation (16) is a constant, and thus has no impact on the value of l^{new}. For efficient calculation, we can rewrite the above equation as

$$l^{\mathrm{new}} = \underset{l \in \{\text{candidate labels}\}}{\arg\min} [\mathrm{Q}^{\mathrm{MDL}}(l_i^{[x]} = l) - \mathrm{Q}^{\mathrm{MDL}}(\mathbf{G} \setminus v_i^{[x]}, \mathcal{L} \setminus l_i^{[x]})]. \tag{17}$$

The second term at the right hand side of Equation (17) is the quality function

for the network subtracting $v_i^{[x]}$ and the partition without considering $l_i^{[x]}$. In this way, many terms which are not related to $v_i^{[x]}$ will be canceled out. Actually, for each candidate label l, we only need to traverse nodes that have links with $v_x^{[i]}$ and traverse communities that have links with the one labeled l. This operation requires a time of $\mathrm{O}(m/n)$, where $m = m^{[1]} + m^{[2]} + \cdots + m^{[s]}$ is the total number of edges, and $n = n^{[1]} + n^{[2]} + \cdots + n^{[r]}$ is the total number of nodes. On average, there are $\mathrm{O}((m/n)^2)$ candidate labels. Thus, updating the label for one node requires a time of $\mathrm{O}((m/n)^3)$. We update labels sequentially for each node and repeat round by round until no decrease of $\mathrm{Q}^{\mathrm{MDL}}$ can be attained. In practice, the number of rounds p is small. In addition, the number of passes q between Phase 1 and Phase 2 is also small. As a result, the overall time complexity of InfoCom is near $\mathrm{O}(pqm^3/n^2)$, where $p, q \ll n$.

§5 Experiments

So far we have proposed an information compression based method for detecting communities in multi-partite multi-relational networks. In this section, we present experiments for testing the performance of our method.

5.1 Synthetic Networks

First, we test our method and compare it with the state-of-the-art techniques in synthetic networks. The basic scheme is as follows. 1) We generate a sequence of synthetic multi-partite 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 the planted communities most accurately. This kind of testing is widely used by other researchers in the community detection field.[1, 16, 17, 42, 43]

As shown in Fig. 5, the 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

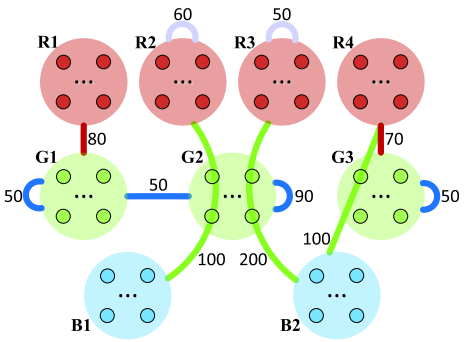


Fig. 5 The link patterns of the communities in the synthetic network. The numbers indicate the numbers of edges.

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. 5 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 link 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 900 edges. Then, we added noise edges randomly. The noise rate increased from 0% to 200% (thus the total number of edges ranges from 900 to 2,700), and the planted communities became 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)[16,44] 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 InfoCom with the following five methods, which cover the state-of-the-art techniques.

- *CompMod*[13]: Detect communities by maximizing composite modularity, which is an integration of the modularity in each subnetwork.
- *NaiveSimp*: Simplify the multi-partite 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 of each edge type).
- *Trans-CN* (A modification of the method proposed in [30]): Transform the multi-partite multi-relational network to weighted uni-partite networks for each node type based on the number of common neighbors. Then detect communities in each uni-partitie network separately.
- *Trans-JD* (A modification of the method proposed in [30]): Transform the multi-partite multi-relational network to weighted uni-partite networks for each node type based on the Jaccard index. Then detect communities in each uni-partitie network separately.
- *MetaFac*[12]: 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.

The results are shown in Fig. 6. It shows that InfoCom algorithm out-

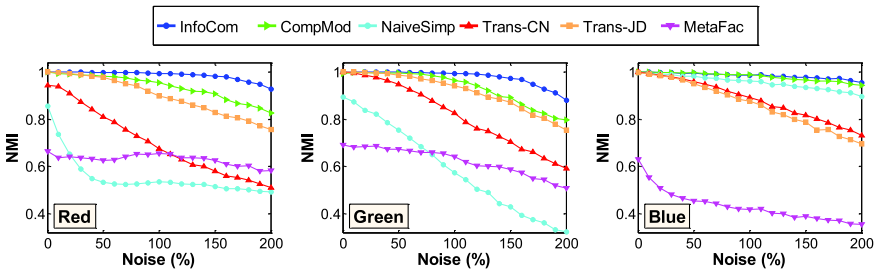


Fig. 6 The NMI values for the red, green, and blue node sets achieved by different methods in the synthetic networks.

Table 2 The runtime of different methods in the synthetic networks. The data is based on a PC equipped with an Intel Core i7-2600 CPU at 3.40 GHz and 32GB physical memory.

Methods	Runtime (seconds)
InfoCom	6.06
ComMod	4.71
NaiveSimp	3.34
Trans-CN	2.61
Trans-JD	2.96
MetaFac	1.27

performs the others by a large margin. It successfully detects 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 0.88. CompMod method performs the second best, accurately detecting the communities in most of the time. However, due to the deficiency in composite modularity definition, this method cannot fully handle communities with many-to-many correspondence. Specifically, it tends to mix communities R2 and R3, so that community G2 only corresponds to one community in the red node set. Consequently, its performance decreases rapidly as the noise increases.

As for other methods, none of them detects the planted communities with 100% accuracy even when the noise is as low as 10%. Specifically, 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 calculate the similarity between nodes. Moreover, as the noise increases, the transformed uni-partite networks of the two methods become so dense that almost all pairs of nodes are connected, posing a great difficulty 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 uni-partite subnetwork of green nodes, there are dense edges both within and between community G1 and G2. Thus, NaiveSimp fails to separate them and take 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. Table 2 compares the runtime of different methods. Although InfoCom is slower than others, it is worthy of waiting such an additional time considering its high accuracy. In summary, this experiment shows that our method is better than the state-of-the-art techniques in detecting the planted communities.

5.2 Digg Network

Second, we apply InfoCom algorithm to a real-world Digg network. Digg is a social news website. Digg users can vote stories (web contents) by "digging" them. 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 multi-partite multi-relational network which contains 7,428 users, 15,117 stories, 166,592 edges representing the friendship between users, and 92,242 edges representing the digging relationship between users and stories. The Digg network can be visualized as a bitmap shown in the left panel of Fig. 7. In this bitmap, there are a user-user plane and a user-story plane, where users and stories are arranged sequentially along the user and story axes. A dot at the user-story plane represents the digging relationship between the corresponding user and story, and a dot at the user-user plane represents the friendship between the corresponding users. We can find that the bitmap is in chaos.

Applying InfoCom algorithm to the Digg network, we detected 30 user communities and 5 story communities. Rearranging users and stories and putting elements within the same communities together, we rebuilt the bitmap. As shown in the right panel of Fig. 7, it is now in an ordered state, implying that the detected communities have similar link patterns. For example, there are dense links within the same user communities; each user community densely links to one or several story communities.

We can check the reasonability of the story communities using the top-

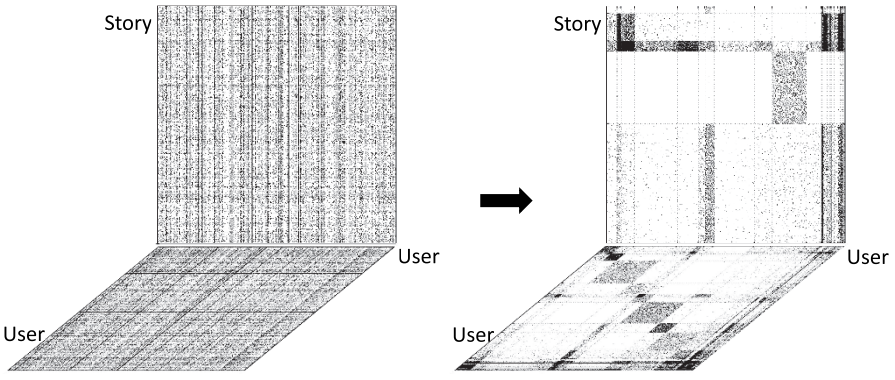


Fig. 7 Visualization of the Digg Network in a Bitmap.

ics associated with stories as a ground truth. In total, there are 10 predesignated topics (business, entertainment, gaming, lifestyle, offbeat, politics, science, sports, technology, and world news), and each story is associated with one of the topics when submitted by a user. We use *averaged F-score*[45] with regard to the topics as a measure for quantifying the detected story communities.

Given a detected story community $\mathbf{V}_\alpha^{[s]}$ and with reference to a story cluster $\mathbf{U}_\beta^{[s]}$ which have the same topic, we define the precision rate as

$$\text{Precision}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}) = \frac{|\mathbf{V}_\alpha^{[s]} \cap \mathbf{U}_\beta^{[s]}|}{|\mathbf{V}_\alpha^{[s]}|}, \tag{18}$$

and the recall rate as

$$\text{Recall}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}) = \frac{|\mathbf{V}_\alpha^{[s]} \cap \mathbf{U}_\beta^{[s]}|}{|\mathbf{U}_\beta^{[s]}|}. \tag{19}$$

The F-score of $\mathbf{V}_\alpha^{[s]}$ on $\mathbf{U}_\beta^{[s]}$ is the harmonic mean of the precision and recall rates:

$$\text{F}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}) = 2 \cdot \frac{\text{Precision}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}) \cdot \text{Recall}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]})}{\text{Precision}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}) + \text{Recall}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]})}. \tag{20}$$

For a detected community $\mathbf{V}_\alpha^{[s]}$, we compute its F-score on each topic cluster and define the maximal obtained as $\mathbf{V}_\alpha^{[s]}$'s F-score. That is,

$$\text{F}(\mathbf{V}_\alpha^{[s]}) = \max_{0 \le \beta \le 9} \text{F}(\mathbf{V}_\alpha^{[s]}, \mathbf{U}_\beta^{[s]}). \tag{21}$$

The averaged F-score of a story partition $\mathcal{L}^{[s]}$ is then calculated as the weighted average of each community's F-score:

$$\text{F}(\mathcal{L}^{[s]}) = \sum_\alpha \frac{|\mathbf{V}_\alpha^{[s]}|}{|\mathbf{V}^{[s]}|} \text{F}(\mathbf{V}_\alpha^{[s]}). \tag{22}$$

As for user communities, we have no ground truth and thus cannot compute averaged F-score. Instead, we analyze the averaged density of user communities. Given a detected user community $\mathbf{V}_\alpha^{[u]}$, its density is defined as

$$\text{D}(\mathbf{V}_\alpha^{[u]}) = \frac{\#\text{ of edges with both ends in } \mathbf{V}_\alpha^{[u]}}{|\mathbf{V}_\alpha^{[u]}| \times (|\mathbf{V}_\alpha^{[u]}| - 1)/2} \tag{23}$$

The averaged density of user partition $\mathcal{L}^{[s]}$ is then calculated as the weighted average of each community's density

$$\text{D}(\mathcal{L}^{[s]}) = \sum_\alpha \frac{|\mathbf{V}_\alpha^{[u]}|}{|\mathbf{V}^{[u]}|} \text{D}(\mathbf{V}_\alpha^{[u]}). \tag{24}$$

We compare InfoCom with various methods. Note that the Digg network is too large to apply Trans-CN, Trans-JD, and MetaFac (the first two have

Table 3 The results obtained by different methods in the Digg network. The runtime is based on a PC equipped with an Intel Core i7-2600 CPU at 3.40 GHz and 32GB physical memory.

Methods	$(c^{[u]}, c^{[s]})$	Avg Density	Avg F-Score	Runtime (mins)
InfoCom	(30, 5)	0.0396	0.530	160.7
ComMod	(132, 149)	0.0398	0.387	123.6
NaiveSimp	(161, 163)	0.0372	0.278	43.4
CESNA°	(58, *n.a.*)	0.0347	*n.a.*	190.0
CODICIL°+BRIM	(37, 37)	0.0385	0.362	59.5
Louvain+BRIM	(56, 56)	0.0504	0.305	25.3

high time complexity, and the last one has high space complexity). To include further competitors, we consider methods which utilize both links and node attributes for community detection.[46–49] In particular, we use a limited version of CESNA[50] and CODICIL[45] (marked as CESNA° and CODICIL° thereafter). These two methods originally take additional information which are beyond links, such as age, nationality, religious belief, education level of a person in a social network for node attributes. In order to apply them to this dataset, we use binary vectors of the links to story nodes[*3] as user nodes' attributes, and then combine with the user-user links for community detection. For CODICIL°, we choose Louvain algorithm[21] for automatically detecting communities in a fused network, since it does not require inputting the number of communities. In addition, CESNA° has some other input parameters, for which we used default values provided by the authors. The output of these two methods is a user partition.[*4] To obtain story partitions, we can further use BRIM algorithm[51] in the user-story subnetwork. BRIM can induce a story partition from a given user partition based on maximizing bipartite modularity, which is a quality function for evaluating community structures in bipartite networks.

From Table 3, we can find that InfoCom achieves the highest F-Score, 40% higher than the second highest score by ComMod. Note that although both InfoCom and ComMod work by optimizing quality functions, the quality functions are based on different principles. InfoCom's quality function is based on compressing the information of the whole multi-partite multi-relational network, whereas ComMod's quality function is based on decomposing the multi-partite multi-relational network into multiple single-relational subnetworks, and comparing each subnetwork with a sub-null model. Thus, the former is formed more from a global angle, whereas the latter is more from a local angle. The difference of the detected communities by these two methods can be seen in Fig. 8. ComMod seeks for communities with one-to-one correspondence, in the sense that most of the user communities densely link to only one story community.

[*3] Only the story nodes which have degrees of more than 10 are considered, since the total number of attributes for these methods should be relatively small.

[*4] CESNA° has a different aim as the one depicted in Section 3 — instead of clustering all of the user nodes into communities, it only clusters part of them which are most likely to constitute communities according to their model criteria.[50] Actually, only 72.9% user nodes are clustered into communities. As a result, we cannot use BRIM algorithm, and thus story community related results for this method are not available in Table 3.

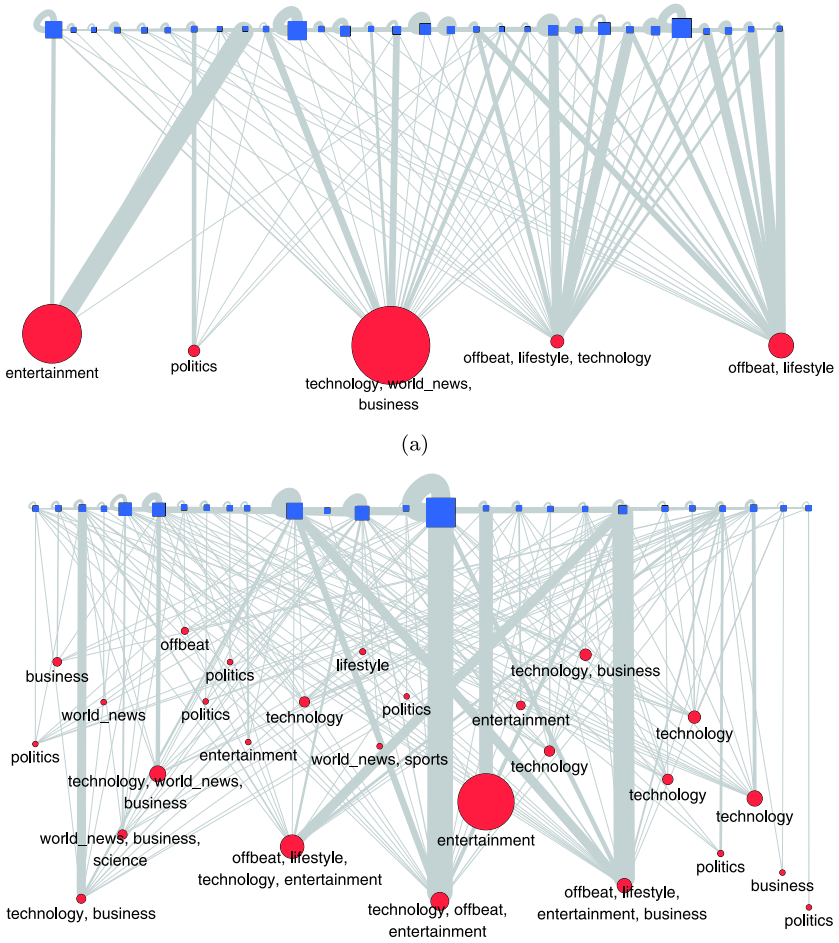


Fig. 8 Visualization of the Digg network at the community level. (a) The communities detected by InfoCom. (b) The communities detected by ComMod (Small communities are omitted. The filter size is set to be 10 and 20 nodes for user and story communities, respectively). The story and user community are depicted by red (round) and blue (rectangle) symbols, respectively. The topics of a story community are exhibited (If there is a single topic which accounts for more than 70% of a community, only this topic is exhibited; otherwise, all topics which account for more than 15% are exhibited).

Consequently, many topic clusters are further divided, so that the number of story communities is near the number of user communities. For example, the politics topic cluster is divided into six smaller communities by ComMod. On the other hand, InfoCom seeks for link-pattern based communities, such that nodes in each detected community have similar link patterns. Thus, a story

community can correspond to multiple user communities, and vice versa. In real world, such a many-to-many correspondence is more reasonable, since a user community can have multiple interests, and they focus on, for example, stories both on technology, world news and business. For this reason, some topic clusters are combined into bigger communities by InfoCom. NaiveSimp has a low F-score too, since it only uses incomplete user-story relationship for analysis. In addition, InfoCom also outperforms CESNA° and CODICIL°, which originally utilize both links and node attributes for community detection.

As for density of user communities, InfoCom achieves a relatively high score which is marginally higher than most of the other methods and comparable to ComMod. Note that the highest density does not necessarily mean the best partition. For example, Louvain algorithm[21] which directly maximizes the number of within-community edges in the user-user network obtains the highest density score. However, its F-score at the story side is not satisfactory. Table 3 also compares the runtime of different methods. Although InfoCom is not the fastest, its runtime is within the range of practical use when dealing with such a large dataset. Therefore, this experiment shows that InfoCom successfully detected reasonable communities that agree with the natural human intuition and perception.

§6 Conclusion

We have extended the information compression based method for detecting communities in multi-partite multi-relational networks. A community is composed of nodes which have similar link patterns. By utilizing such similarity, we convert the problem of community detection to a problem of finding an efficient compression of the network's structure. Specifically, based on the MDL principle which accounts for the best compression of network structure data, we propose a quality function for evaluating partitions of a multi-partite multi-relational network into communities, and develop a heuristic algorithm called InfoCom for optimizing the quality function. Our method overcomes the limitations of existing methods and is applicable to communities of many-to-many correspondence. In addition, our method does not require a priori knowledge about the numbers of communities. Our experiments in both synthetic and real-world networks demonstrate that InfoCom outperforms the state-of-the-art techniques. As part of our future work, we plan to apply our method to giga-scale networks using high-performance computing resources, such as multi-cores, GPUs, and clusters.

Acknowledgements

We gratefully thank Mr. Pen-Lin Chang for collecting the Digg data. This work was partly supported by the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science and Technology (CREST) research project, and the National Natural Science Foundation of China (Grant No. 61203154).

References

1) Girvan, M. and Newman, M. E. J., "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA, 99*, pp. 7821–7826, 2002.
2) Flake, G. W., Lawrence, S., Giles, C. L. and Coetzee, F. M., "Self-organization and identification of web communities," *IEEE Computer, 35, 3*, pp. 66–70, 2002.
3) Dourisboure, Y., Geraci, F. and Pellegrini, M., "Extraction and classification of dense communities in the web," in *Proc. of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada), pp. 461–470, May 2007.
4) Guimerà, R. and Amaral, L. A. N., "Functional cartography of complex metabolic networks," *Nature, 433*, pp. 895–900, 2005.
5) Palla, G. Derényi, I. Farkas, I. and Vicsek, T., "Uncovering the overlapping community structure of complex networks in nature and society," *Nature, 435, 7043*, pp. 814–818, 2005.
6) Fortunato, S., "Community detection in graphs," *Physics Reports, 486*, pp. 75–174, 2010.
7) Gulbahce, N. and Lehmann, S., "The art of community detection," *BioEssays, 30, 10*, pp. 934–938, 2008.
8) Newman, M. E. J., *Networks: An Introduction*, New York, Oxford University Press, 2010.
9) Lancichinetti, A., Radicchi, F., Ramasco, J. J. and Fortunato, S., "Finding statistically significant communities in networks," *PloS one, 6, 4, e18961*, 2011.
10) Sun, Y., Yu, Y. and Han, J., "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France), pp. 797–806, Aug 2009.
11) Comar, P. M., Tan, P. N. and Jain, A. K., "A framework for joint community detection across multiple related networks," *Neurocomputing, 76, 1* pp. 93–104, 2012.
12) Lin, Y. R., Sun, J., Castro, P., Konuru, R., Sundaram, H. and Kelliher, A., "Metafac: community discovery via relational hypergraph factorization," in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France), pp. 527–535, Aug 2009.
13) Liu, X., Liu, W., Murata, T. and Wakita, K., "A framework for community detection in heterogeneous multi-relational networks," *Advances in Complex Systems, 17, 1450018*, pp. 1–21, 2014.
14) Rissanen, J., "Modelling by shortest data description," *Automatica, 14, 5*, pp. 465–471, 1978.
15) Newman, M. E. J. "Communities, modules and large-scale structure in networks," *Nature Physics, 8, 1*, pp. 25–31, 2012.
16) Danon, L., Duch, J., D.-Guilera, A. and Arenas, A. "Comparing community structure identification," *J. Stat. Mech., P09008*, 2005.
17) Lancichinetti, A. and Fortunato, S., "Community detection algorithms: a comparative analysis," *Phys. Rev. E*, *80, 056117*, 2009.
18) Orman, G. K., Labatut, V. and Cherifi, H., "Comparative evaluation of community detection algorithms: a topological approach," *J. Stat. Mech., P08001*, 2012.

19) Newman, M. E. J., "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, 69, 066133*, 2004.

20) Clauset, A., Newman, M. E. J. and Moore, C., "Finding community structure in very large networks," *Phys. Rev. E, 7, 066111*, 2004.

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

22) Liu, X. and Murata, T., "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Physica A, 389, 7*, pp. 1493–1500, 2010.

23) Newman, M. E. J., "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, 74, 036104*, 2006.

24) Shen, H. and Cheng, X., "Spectral methods for the detection of network community structure: a comparative analysis," *J. Stat. Mech., P10020*, 2010.

25) Ball, B. Karrer, B. and Newman, M. E. J., "Efficient and principled method for detecting communities in networks," *Phys. Rev. E, 84, 036103*, 2011.

26) Karrer, B. and Newman, M. E. J., "Stochastic blockmodels and community structure in networks," *Phys. Rev. E, 83, 016107*, 2011.

27) 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*, pp. 876–878, 2010.

28) Tang, L., Liu, H. and Zhang, J., "Identifying evolving groups in dynamic multimode networks," *IEEE Transactions on Knowledge and Data Engineering 24, 1*, pp. 72–85, 2012.

29) Tang, L., Wang, X. and Liu, H., "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery, 25, 1*, pp. 1–33, 2012.

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

31) Dhillon, I. S., Mallela, S. and Modha, D. S., "Information-theoretic coclustering," in *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC, USA), pp. 89–98, Aug 2003.

32) Rosvall, M. and Bergstrom, C. T., "An information-theoretic framework for resolving community structure in complex networks," *Proc. Natl. Acad. Sci. USA, 104, 18*, pp. 7327–7331, 2007.

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

34) Liu, X. and Murata, T., "Detecting communities in k-partite k-uniform (hyper) networks," *Journal of Computer Science and Technology, 26, 5*, pp. 778–791, 2011.

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

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

37) Rissanen, J. "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics, 11, 2*, pp. 416–431, 1983.

38) Waltman, L. and van Eck, N. J., "A smart local moving algorithm for largescale modularity-based community detection," *The European Physical Journal B, 86, 11, 471*, 2013.

39) Rotta, R. and Noack, A., "Multilevel local search algorithms for modularity clustering," *ACM Journal of Experimental Algorithmics, 16, 2, 2.3*, 2011.

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

41) Arenas, A., Duch, J., Fernández, A. and Gómez, S., "Size reduction of complex networks preserving modularity," *New Journal of Physics, 9 176*, 2007.

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

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

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

45) Ruan, Y., Fuhry, D. and Parthasarathy, S., "Efficient community detection in large networks using content and links," in *Proc. of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil), pp. 1089–1098, May 2013.

46) Günnemann, S., Boden, B. and Seidl, T., "Db-csc: a density-based approach for subspace clustering in graphs with feature vectors," in *Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (Athens, Greece), pp. 565–580, Sep 2011.

47) Günnemann, S., Färber, I., Boden, B. and Seidl, T. "Gamer: a synthesis of subspace clustering and dense subgraph mining," *Knowledge and information systems, 40, 2*, pp. 243–278, 2013.

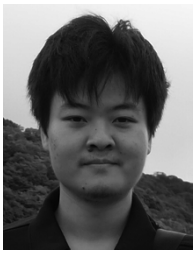
48) Günnemann, S., Boden, B., Färber, I. and Seidl, T., "Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors," in *Proc. of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Gold Coast, Australia), pp. 261–275, Apr 2013.

49) Boden, B., Ester, M. and Seidl, T., "Density-based subspace clustering in heterogeneous networks," in *Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (Nancy, France), pp. 149–164, Sep 2014.

50) Yang, J., McAuley, J. and Leskovec, J., "Community detection in networks with node attributes," in *Proc. of the 13th IEEE International Conference on Data Mining*, (Dallas, TX, USA), pp. 1151–1156, Dec 2013.

51) Barber, M. J., "Modularity and community detection in bipartite networks," *Phys. Rev. E, 76, 066102*, 2007.

Xin Liu, Dr. Engineering: He is a researcher at the Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He holds B.Sc.(Computing and Information Science), M.Sc.(Computer Science), and doctor's degree (Computer Science) from Wuhan University of Technology, Wuhan University, and Tokyo Institute of Technology, respectively. He also works as an associate professor at the Department of Mathematics, School of Science, Wuhan University of Technology. His research focuses mostly on graph mining and social network analysis.



Weichu Liu: He received his B.Sc. degree in Computer Science from The University of Electronic Science and Technology of China and his M.Sc. degree in Computer Science from Tokyo Institute of Technology. His research interests include complex network analysis and data mining. Currently he works as a PaaS (Platform as a Service) engineer at Rakuten Inc., an e-commerce company based in Tokyo.

Tsuyoshi Murata, Dr. Engineering: He is an associate professor in the Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He obtained his doctor's degree in Computer Science at Tokyo Institute of Technology in 1997, on the topic of Machine Discovery of Geometrical Theorems. At Tokyo Institute of Technology, he conducts research on Web mining and social network analysis.

Ken Wakita, Dr. Science: He is an associate professor in the Department of Mathematical and Computing Sciences, Graduate School of Information Science and Engineering, Tokyo Institute of Technology. His current interests include social network analysis, information visualization, and compiler construction. He received his doctor's degree in science at The University of Tokyo in 1997. He is a member of ACM and IEEE computer society.