Chapter 4
Discovering Communities in Multi-relational Networks

Zhiang Wu, Zhan Bu, Jie Cao and Yi Zhuang

Abstract Multi-relational networks (in short as MRNs) refer to such networks including one-typed nodes but associated with each other in poly-relations. MRNs are prevalent in the real world. For example, interactions in social networks include various kinds of information diffusion: email exchange, instant messaging services and so on. Community detection is a long-standing yet very difficult task in social network analysis, especially when meeting MRNs. This chapter gradually explores the research into discovering communities from MRNs. It begins by introducing the generalized modularity of the MRN, which paves the way for applying modularity optimization-based community detection methods on MRNs. However, the mainstream methods for discovering communities on MRNs are to integrate information from multiple dimensions. The existing integration methods fall into four categories: network integration, utility integration, feature integration, and partition integration. Learning or ranking the weight for each relation in MRN constitutes building blocks of network, utility and feature integrations. Thus, we turn our attention into several co-ranking frameworks on MRNs. We then discuss two different kinds of partition integration strategies, including the frequent pattern mining based method and the consensus clustering based method. Finally, for the purpose of conducting performance validation, we present several techniques for constructing the MRN based on both multivariate data and forum data.

Z. Wu (✉) · Z. Bu · J. Cao
National Center for International Joint Research on E-Business Information Processing (NECC), School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China
e-mail: zawuster@gmail.com

Y. Zhuang
College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China
4.1 Introduction

Multi-relational networks (MRNs), being composed of one-typed nodes but multi-typed relations, have been found ubiquitous in the real world. For example, in Twitter, the multi-relations among users contain followers or followees, retweeting tweets, publishing relevant tweets with others, and so on. There are also various types of relations among users in social networks, including friendship, contact, co-subscription or co-tagging to the same resource, co-contact with the third one, and so on. Another important reason attracting a great deal of focus to MRNs is that the MRN is closely related to the time-dependent network [18]. That is, if taking every snapshot as a network in one relation, the time-dependent network becomes a MRN. Relations in MRNs can be either explicit or implicit [3]. Explicit relations directly reflect the various interactions in reality, but implicit relations are inferred from the available data to reflect different interesting qualities of the interactions.

Some studies target at establishing theoretical basis for multi-relational network analysis, such as algebra operations [22] and analytical measures extensions [2, 3]. Community discovery in MRNs has become one of the prevailing topic among studies on multi-relational network analysis. Similar to community detection methods with global models in single-relational networks, Mucha et al. [18] proposed the generalized modularity ($Q$ function), and aimed to optimize this criterion defined over the network partition. However, most of studies [26, 32, 34] attempted to integrate information from multiple dimensions to discover the shared community structure across multiple network dimensions. According to the four components involved in a network, the existing integration methods can be summarized as four categories: network integration, utility integration, feature integration, and partition integration [26]. After any integration strategy applied on a MRN, various community detection methods (such as spectral clustering, block models, latent space approach, and modularity optimization) will be applicable on the MRN. An unified view for integration and then community detection has been given in [26, 28].

The goal of three integration strategies, i.e., network integration, utility integration, and feature integration, is very similar. That is, they target at transforming a MRN to a SRN, by integrating topological, utility matrices, and structural features from multiple relations with average or biased weights, respectively. So, how to learn/rank the weight for every relation becomes an important problem during the integration process. In this chapter, we discuss several co-ranking frameworks on the multi-relational data/network, which differentiates with the unified view introduced in [26, 28]. As for partition integration, we review two different methods including the frequent pattern mining based method and the consensus clustering based method.

The remainder of this chapter is organized as follows. In Sect. 4.2, we formulate the problem of community discovery on multi-relational networks. In Sect. 4.3, we introduce the concept of generalized modularity, which lays the foundation for modularity optimization based community detection. In Sect. 4.4, we first overview network/utility/feature integration strategies, and thus highlight co-ranking techniques
that serve as building blocks for these integration strategies. In Sect. 4.5, we present two kinds of partition integration methods. We discuss some issues for conducting performance validation on MRNs in Sect. 4.6, and finally conclude this chapter in Sect. 4.7.

4.2 Problem Formulation

This section provides preliminary knowledge of the MRN and community discovery on it, which serves as a consolidated reference for the following sections. A single-relational network is usually represented as an adjacency matrix that is also known as a two-way tensor because it has two dimensions. Stated in this way, an MRN can be represented as a three-way tensor consisting of a series of matrix “slice”. We formally define the MRN as follows:

Definition 4.1 (Multi-relational Network) A $m$-relational network is defined as a set of graphs $M = (V, \mathbb{E} = \{E_1, \ldots, E_m \subseteq (V \times V)\})$, where $V$ denotes the node set with $n$ elements, and $\mathbb{E}$ is the $m$-relational edge set. Let $A \in \{0, 1\}^{n \times n \times m}$ be a three-way tensor, then

$$A^d_{i,j} = \begin{cases} 
1 & \text{if } (i, j) \in E_d : 1 \leq d \leq m \\
0 & \text{otherwise} 
\end{cases} \quad (4.1)$$

In this formulation, $A$ represents the primary structure by which every adjacency matrix “slice” $A^d \in \{0, 1\}^{n \times n}$ is indexed. An MRN actually contains homogeneous nodes but heterogeneous relations, which is different with the multi-mode network or the heterogeneous network [28]. Figure 4.1 illustrates the difference between the MRN and the multi-mode network. Although much research has been done on discovering communities from static or dynamic multi-mode networks [24, 27, 28], the scope of this chapter is limited to community detection from MRNs.

Since nodes in an MRN are homogeneous, the problem of discovering communities on MRNs is similar to that on SRNs. That is, community detection aims to find a good $K$-way partition $\mathcal{P} = \{C_1, \ldots, C_K\}$, where $C_k$ is the $k$th community, and $C_1 \cup \cdots \cup C_K \subseteq V$. $K$ can either be given in advance or determined by the community detection algorithm itself. For a crisp partition, we have an additional requirement: $C_k \cap C_{k'} = \emptyset \; \forall \; k \neq k'$. However, for an overlapping or fuzzy partition, overlapping communities can be represented as a membership matrix $U = [u_{i,k}], i = 1, \ldots, n, k = 1, \ldots, K$, where $0 \leq u_{i,k} \leq 1$ denotes the membership that node $i$ belongs to $C_k$. If node $i$ belongs to only one community, $u_{i,k} = 1$, and it clearly follows that $\sum_{k=1}^{K} u_{i,k} = 1$ for all $1 \leq i \leq n$. 
4.3 Generalized Modularity Optimization

The choice of null model is a crucial consideration for modularity definition and thus for community detection. In the early literature, many null models have been proposed for modularity definition in the SRNs, among which, Newman-Girvan’s [19] is the most used and best known one. The basic idea behind Newman-Girvan’s null model is that a random graph is not expected to have a community structure, so the possible existence of communities is revealed by the comparison between the intra-community edge weight and that expected at random. Other null models for the SRNs along this line can be found in [1, 11, 16]. However, such null models have not been available for MRNs.

Figure 4.2 shows such an example with a typical MRN defined by coupling multiple adjacency matrices, where the connections encoded by the network “slices” are flexible; they can represent variations across time, variations across different types of connections, or even community detection of the same network at different scales. Let $A^d_{i,j}$ represent the intra-slice coupling that connect node $i$ and node $j$ in slice $d$, and $C^d_{i,r}$ indicate the inter-slice coupling that connect node $i$ in slice $r$ to itself in slice $d$. As these inter-slice couplings are either present or absent by definition, when they do fall inside communities, their contribution in count of intra-community edges exactly cancels that expected at random. Therefore, the usual null models fails to provide any contribution from these inter-slice couplings. In contrast, by formulating a null model in terms of stability of communities under Laplacian dynamics, one can derive a principled generalization of community detection to MRNs [18].
4.3.1 Laplacian Dynamics Formalism

The Laplacian dynamics formalism, which has been recently developed by Lambiotte et al. [13], is to rederive network modularity from the continuous-time normalized Laplacian dynamics \( \dot{p}_i = \sum_j \frac{1}{k_j} A_{i,j} p_j - p_i \) on a unipartite, undirected network defined by the adjacency matrix components \( A_{i,j} \) with node strengths \( k_i = \sum_j A_{i,j} \). Note that there is a steady state given by \( p_j^* = k_j/(2m) \), where \( 2m = \sum_i k_i = \sum_{i,j} A_{i,j} \), describes the total strength in the network. So the stability of communities under such dynamics can be measured by directly comparing the joint probability at stationarity of independent appearances at nodes \( i \) and \( j \) with the linear approximate map from node \( j \) to node \( i \). Under the guidance of this direction, Lambiotte et al. quantified a measure of the stability \( R(t) \) of a specified partition of the network into communities using the probability that a random walker remains within the same community after time \( t \), in statistically steady conditions, relative to that expected under independence. Given the operator \( L_{i,j} = A_{i,j}/k_j - \delta_{i,j} \) of the dynamics, where \( \delta_{i,j} \) is the Kronecker delta, the stability \( R(t) \) is defined as following:

\[
R(t) = \sum_{i,j} \left[ (e^{tL})_{i,j} p_j^* - p_i^* p_j^* \right] \delta(g_i, g_j), \tag{4.2}
\]

where \( p_i^* p_j^* \) denotes the contribution from an independence assumption. Expanding the matrix exponential in Eq. (4.2) to first-order in \( t \), we have \( (e^{tL})_{i,j} \simeq \delta_{i,j} + t L_{i,j} \).
As \( \delta_{i,j} \) factors always contribute to the sum, \( R(t) \) can be directly yielded to the quality function \( Q(t) = \frac{1}{2m} \sum_{i,j} [r A_{i,j} - \frac{k_{i,j}}{2m}] \delta(g_i, g_j) \). When \( t = 1 \), the resulting quality function reduces to Newman-Girvan modularity. Moreover, if both sides of the equation are divided by \( t \), the quality can be written in the usual form: 
\[
Q = \frac{1}{2m} \sum_{i,j} [A_{i,j} - \gamma \frac{k_{i,j}}{2m}] \delta(g_i, g_j),
\]
with the resolution parameter \( \gamma = 1/t \). Hence, the stability of the community partition relative to that expected under independence provides a natural definition for the null model employed in the quality function.

### 4.3.2 Generalized Laplacian Dynamics

Along the above line, Mucha et al. [18] developed a generalized framework of network quality functions that allowed us to study the community structure of MRNs, which are combinations of individual networks coupled through links that connect each node in one network slice to itself in other slices. Without loss of generality, we restricted our attention to undirected network slices (i.e., \( A_{d,i,j} = A_{d,j,i} \)) and undirected couplings (i.e., \( C_{d,i,j} = C_{d,j,i} \)). Notating the strengths of each node individually in each slice by \( k_{i,d} \), and across slices by \( c_{i,d} \). Thus, the multi-slice strength of the node is given by \( \kappa_{i,d} = k_{i,d} + c_{i,d} \), and the continuous-time Laplacian dynamics is defined as

\[
\dot{p}_{i,d} = \sum_{j,r} \left( A_{d,i,j} \delta_{d,r} + \delta_{i,j} C_{d,j} \right) \frac{p_{j,r}}{\kappa_{j,r}} - p_{i,d},
\]

which respects the intra-slice nature of \( A_{d,i,j} \) and the inter-slice couplings of \( C_{d,j} \). The steady state in such case is \( p_{j,r}^* = \kappa_{j,r} / \sum_{j,r} \kappa_{j,r} \). Thus, the associated multi-slice null model can be specified by the probability \( \rho_{i,d|j,r} \) of sampling node-slice \((i, d)\) conditional on whether the multi-slice structure allows one to step from node-slice \((j, r)\) to node-slice \((i, d)\).

\[
\rho_{i,d|j,r} p_{j,r}^* = \left[ \frac{k_{i,d} k_{j,r} \delta_{d,r}}{2m_d \kappa_{j,r}} + \frac{C_{d,r} \delta_{i,j}}{c_{j,r} \kappa_{j,r}} \right] \sum_{j,r} \frac{\kappa_{j,r}}{\kappa_{j,r}},
\]

where \( m_d = \sum_i k_{i,d} \). That is, the conditional probability of stepping from \((j, r)\) to \((i, d)\) along an inter-slice coupling is nonzero if \( i = j \), and it is proportional to the probability \( C_{d,r} / k_{i,r} \) of selecting the precise inter-slice link that connects to slice \( d \). Subtracting this conditional joint probability from the linear approximation of the exponential describing the Laplacian dynamics on MRNs, a multi-slice generalization of modularity can be obtained as follows.
\[
Q_{\text{MRN}} = \frac{1}{\sum_{j,r} \sum_{i,j,d,r} \left( A_{i,j,d} - \gamma_d \frac{k_i,d k_j,d}{2m_d} \right) \delta_{d,r} + \delta_{i,j} C_{d,r}^{d,r}} \delta(g_{i,d}, g_{j,r}).
\]  (4.5)

Some notable remarks for Eq. (4.5) should be highlighted. First, the corresponding resolution parameter for the inter-slice couplings is absorbed into the magnitude of the elements of \( C_{d,r}^{d,r} \), which is a binary value in \{0, w\}. \( w = 0 \) indicates there is no benefit from extending communities across slices, and therefore the optimal partition is obtained from independent optimization of the corresponding quality function in each slice. Otherwise, when \( w \) becomes sufficiently large, the quality-optimizing partitions force the community assignment of a node to remain the same across all slices in which that node appears, and the multi-slice quality reduces to a difference between the adjacency matrix summed over the contributions from the individual slices and the sum over the separate single-slice null models.

Second, a re-weighting technique was used in conditional probabilities, which allows for different resolutions \( \gamma_d \) in each slice. In the absence of such a re-weighting in the interpretation of the stability of the partition, with \( \gamma_d = \gamma \) for all \( d \), the corresponding prefactor on \( C_{d,r}^{d,r} \) absorbed above is \((1 - \gamma)\). Imposing the choice \( \gamma = 1 \) then recovered the usual interpretation of modularity as a count of the total weight of intra-slice edges minus the weight expected at random, and the specified deterministic \( C_{d,r}^{d,r} \) contribution was dropped out entirely.

Community discovery in MRNs can then proceed using many of the same computational heuristics that are currently available for single-relational networks. During this process, one may exert special caution about the resolution of communities and the likelihood of complex quality landscapes that necessitate caution in interpreting results on real-world networks.

### 4.4 Co-Ranking Frameworks

With multi-typed interactions, the community structures hidden in MRNs can be complicated. Integrating information from multiple dimensions for community discovery has become the dominant method [6, 7, 32, 34]. In this section, we first overview three integration strategies, i.e., network integration, utility integration, and feature integration, and thus point out that the key issue in these integration strategies is how to rank the weight for every relation. We then introduce two kinds of co-ranking frameworks (i.e., MultiRank and MutuRank) as a complemental technique to integration methods.
4.4.1 Integration Methods: An Overview

Without loss of generality, let a three-way tensor \( S = [s_{i,j,d}] \), \( 1 \leq i, j \leq n, 1 \leq d \leq m \) denote the information/object to be integrated. For network integration, \( S = A \) is the \( m \)-relational edge set. For utility integration, \( S \) consists of \( m \) utility matrices, which is equivalent to optimizing the objective function over all types of interactions simultaneously [26]. For feature integration, \( S \) denotes structural features associated with nodes, commonly extracted by overlapping community detection methods from each relation. If we further let \( \overrightarrow{p} = [p_1, p_2, \ldots, p_n] \) and \( \overrightarrow{q} = [q_1, q_2, \ldots, q_m] \) denote vectors representing the importance weights of nodes and relations on an MRN, respectively. The synthesized weight matrix \( W = [w_{i,j}] \) can be calculated as follows.

\[
w_{i,j} = \sum_{d=1}^{m} q_d \cdot s_{i,j,d}. \tag{4.6}
\]

The key task in the integration is to determine the weight vector of relations \( \overrightarrow{q} \). The simplest way used in [6, 7, 26] modeled \( \overrightarrow{q} \) as an uniform distribution, i.e., \( q_d = \frac{1}{m} \), which obviously failed to distinguish different roles played by various relations. If the network is single-relational, several well-known algorithms such as HITS [12] and PageRank [21] can be applied to rank the importance of nodes, i.e., to compute \( \overrightarrow{p} \). As the network become multi-relational, nodes and relations exert mutual influences to each other, and people need to co-rank both nodes and relations simultaneously. To address this challenge, two kinds of co-ranking algorithms MultiRank [20] and MutuRank [32, 34] have been presented. Before diving into the algorithmic details, we have to introduce some basic operations and definitions. Let \( \mathbb{R} \) be the real field. We define two vectors \( A\overrightarrow{p} \overrightarrow{q} \in \mathbb{R}^n \) and \( A\overrightarrow{p}' \overrightarrow{p}' \in \mathbb{R}^m \) as

\[
\begin{align*}
(A\overrightarrow{p} \overrightarrow{q})_i &= \sum_{j=1}^{n} \sum_{d=1}^{m} a_{i,j,d} \cdot p_j \cdot q_d, \quad i = 1, 2, \ldots, n, \tag{4.7} \\
(A\overrightarrow{p}' \overrightarrow{p}')_d &= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j,d} \cdot p_i \cdot p'_j, \quad d = 1, 2, \ldots, m. \tag{4.8}
\end{align*}
\]

As we imagine a random walk applied on an MRN, we can construct two transition probability tensors \( O = [o_{i,j,d}] \) and \( R = [r_{i,j,d}] \) with respect to objects and relations by normalizing the entries of \( A \) as follows [20]:

\[
o_{i,j,d} = \frac{a_{i,j,d}}{\sum_{l=1}^{n} a_{i,l,d}}, \quad r_{i,j,d} = \frac{a_{i,j,d}}{\sum_{e=1}^{m} a_{i,j,e}}. \tag{4.9}
\]

Figure 4.3a illustrates the construction of \( O \) and \( R \) based upon \( A \). To be specific, tensors \( O \) and \( R \) have same orders with \( A \), where \( o_{i,j,d} \) is normalized by the \( i \)th row in \( d \)th relation and \( r_{i,j,d} \) is normalized by the vertical line fixed by \( i \) and \( j \). Let \( X_t \)
and $Y_t$ be random variables referring to visit at any particular node and to use any particular relation respectively at the time $t$. We have:

$$o_{i,j,d} = \text{Prob} [X_t = i \mid X_{t-1} = j, Y_t = d], \quad (4.10)$$
$$r_{i,j,d} = \text{Prob} [Y_t = d \mid X_t = i, X_{t-1} = j]. \quad (4.11)$$

Clearly, the sequence of random variables $(X_t, Y_t : t = 0, 1, \ldots)$ is a Markov chain. The co-ranking algorithms try to compute the following probabilities with respect to two transition probability tensors $\mathcal{O}$ and $\mathcal{R}$.

$$\text{Prob} [X_t = i] = \sum_{j=1}^{n} \sum_{d=1}^{m} o_{i,j,d} \cdot \text{Prob} [X_{t-1} = j, Y_t = d], \quad (4.12)$$
$$\text{Prob} [Y_t = d] = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{i,j,d} \cdot \text{Prob} [X_t = i, X_{t-1} = j]. \quad (4.13)$$

Therefore, $\vec{p}$ and $\vec{q}$ are equilibrium/stationary distributions of nodes and relations. The most important property of the stationary distributions is that if the network $M$ is non-bipartite, then the distributions of $X_t$ and $Y_t$ tend to stationary distributions, as $t \rightarrow \infty$. Formally, we have:

$$p_i = \lim_{t \rightarrow \infty} \text{Prob} (X_t = i), \quad q_d = \lim_{t \rightarrow \infty} \text{Prob} (Y_t = d).$$

From above analysis, we can summarize that to compute two joint probability distributions $\text{Prob} [X_{t-1} = j, Y_t = d]$ and $\text{Prob} [X_t = i, X_{t-1} = j]$ becomes the key operation for determining the stationary distributions $\vec{p}$ and $\vec{q}$. For this, MultiRank and MutuRank have presented different solutions, which will be introduced in the following sub-sections.

---

**Fig. 4.3** Illustration of some variables in MutiRank and MutuRank. **a** Tensors $\mathcal{O}$ and $\mathcal{R}$. **b** Conditional probabilities.
4.4.2 MultiRank Algorithm

The important assumption in MultiRank [20] is that two random variables (i.e., $X_t$ and $Y_t$) are independent from each other. Two joint probability distributions can therefore be modeled as a product form of individual probability distributions. More precisely, MultiRank assumed that

$$\text{Prob}[X_{t-1} = j, Y_t = d] = \text{Prob}[X_{t-1} = j] \cdot \text{Prob}[Y_t = d], \quad (4.14)$$

$$\text{Prob}[X_t = i, X_{t-1} = j] = \text{Prob}[X_{t-1} = j] \cdot \text{Prob}[X_t = i]. \quad (4.15)$$

Under this assumption, Eqs. (4.12) and (4.13) can be written as the iteration form with respect to $t$.

$$p^{(t+1)}_i = \sum_{j=1}^{n} \sum_{d=1}^{m} o_{i,j,d} \cdot p^{(t)}_j \cdot q^{(t)}_d, \quad (4.16)$$

$$q^{(t+1)}_d = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{i,j,d} \cdot p^{(t)}_i \cdot p^{(t)}_j. \quad (4.17)$$

We can see from Eqs. (4.16) and (4.17) that the MultiRank makes full use of the mutual influence between relations and nodes. Specifically, the mutual feedback here means that: (i) the importance of a relation depends on the probability distribution of nodes and the importance of other relations, i.e., a relation, selected by high-weight nodes with high probabilities, deserves high weight itself; (ii) the importance of a node depends on the probability distribution of relations and its neighbors’ importance, i.e., a node, linked by high-weight nodes with strong and high-weight relations, deserves high-weight. By using the tensor operation shown in Eqs. (4.7) and (4.8), we can represent Eqs. (4.16) and (4.17) in a concise manner, i.e., matrix notations.

$$\overrightarrow{p^{(t+1)}} = \mathcal{O} \overrightarrow{p^{(t)}} \overrightarrow{q^{(t)}}, \quad \overrightarrow{q^{(t+1)}} = \mathcal{R} \overrightarrow{p^{(t)}} \overrightarrow{p^{(t)}}. \quad (4.18)$$

According to Eq. (4.18), an iterative algorithm can naturally be designed to compute $\overrightarrow{p}$ and $\overrightarrow{q}$ for the MultiRank. It starts by assigning random values as $\overrightarrow{p^{(0)}}$ and $\overrightarrow{q^{(0)}}$, and then applies iterative computation based on Eq. (4.18) until the Frobenius norm converges to a tolerance value.

4.4.3 MutuRank Algorithm

The assumption that $X_t$ and $Y_t$ are independent from each other in MultiRank might be too strong. However, the distributions of nodes and relations are actually coupled
together. We therefore propose to use the conditional probability for modeling two joint probability distributions [32, 34].

\[
\text{Prob}[X_{t-1} = j, Y_t = d] = \text{Prob}[X_{t-1} = j] \cdot \text{Prob}[Y_t = d|X_{t-1} = j], \quad (4.19)
\]
\[
\text{Prob}[X_t = i, X_{t-1} = j] = \text{Prob}[X_{t-1} = j] \cdot \text{Prob}[X_t = i|X_{t-1} = j]. \quad (4.20)
\]

Given a node \( j \), the probabilities for selecting the \( d \)th relation, and for transiting from \( j \) to its neighbor \( i \) are as follows. Figure 4.3b shows the computations of two conditional probabilities in three-way tensor.

\[
\text{Prob}[Y_t = d|X_{t-1} = j] = \text{Prob}[d|j] = \frac{q_d \cdot \sum_{l=1}^{n} a_{j,l,d}}{\sum_{l=1}^{n} \sum_{d=1}^{m} q_d \cdot a_{j,l,d}}, \quad (4.21)
\]
\[
\text{Prob}[X_t = i|X_{t-1} = j] = \text{Prob}[i|j] = \frac{\sum_{d=1}^{m} q_d \cdot a_{j,i,d}}{\sum_{l=1}^{n} \sum_{d=1}^{m} q_d \cdot a_{j,l,d}}. \quad (4.22)
\]

Therefore, by using conditional probabilities, Eqs. (4.12) and (4.13) become:

\[
p^{(t+1)}_i = \sum_{j=1}^{n} \sum_{l=1}^{m} p^{(t)}_j \cdot a_{i,j,d} \cdot \frac{q^{(t)}_d \cdot \sum_{l=1}^{n} a_{j,l,d}}{\sum_{l=1}^{n} \sum_{d=1}^{m} q^{(t)}_d \cdot a_{j,l,d}}, \quad (4.23)
\]
\[
q^{(t+1)}_d = \sum_{i=1}^{n} \sum_{j=1}^{n} p^{(t)}_j \cdot r_{i,j,d} \cdot \frac{\sum_{d=1}^{m} q^{(t)}_d \cdot a_{j,i,d}}{\sum_{l=1}^{n} \sum_{d=1}^{m} q^{(t)}_d \cdot a_{j,l,d}}. \quad (4.24)
\]

The iterative form of MutuRank shown in Eqs. (4.23) and (4.24) seemed somewhat complicated. However, similar to any random walk model, we also can represent MutuRank using concise matrix notations. To this end, we have to define two auxiliary matrices \( \mathbf{V} = [\overrightarrow{V}_j]_{m \times 1} \) and \( \mathbf{U} = [\overrightarrow{U}_j]_{n \times 1}, j = 1, \ldots, n \). \( \mathbf{V} \) and \( \mathbf{U} \) are \( n \times m \) and \( n \times m \) dimensional, and they are represented by \( nm \times 1 \) dimensional vectors \( \overrightarrow{V}_j \) and \( n \times 1 \) dimensional vectors \( \overrightarrow{U}_j \), respectively. \( \overrightarrow{V}_j = [v_{j,d}] \) and \( \overrightarrow{U}_j = [u_{j,i}] \) are defined as:

\[
v_{j,d} = q_d \cdot \sum_{l=1}^{n} s_{j,l,d}, \quad u_{j,i} = \sum_{d=1}^{m} q_d \cdot s_{j,i,d}.
\]

If we let \( \mathbf{V} \) and \( \mathbf{U} \) be row-normalized, we have:

\[
\text{Prob}(Y_t = d|X_{t-1} = j) = v_{j,d}, \quad \text{Prob}(X_t = i|X_{t-1} = j) = u_{j,i}.
\]

Under the tensor operations, Eqs. (4.23) and (4.24) can be simplified as:

\[
\overrightarrow{p^{(t+1)}} = \mathcal{O}\overrightarrow{p^{(t)}}\mathbf{V}, \quad \overrightarrow{q^{(t+1)}} = \mathcal{R}\overrightarrow{p^{(t)}}\mathbf{U}. \quad (4.25)
\]
Obviously, $\vec{p}$ and $\vec{q}$ of the MutuRank also can be computed by an iterative algorithm being similar to MultiRank.

4.5 Partition Integration

If we consider a MRN as $m$ independent single-relational network, we can utilize any community detection method on every slice. As the community partition of each slice is ready, partition integration takes effect, and it targets at assembling multiple community partitions as a consensus single community partition. In this section, we introduce two different kinds of method for partition integration.

4.5.1 Frequent Itemsets Mining Based Method

Berlingerio et al. [4] devised a novel algorithm named frequent pAttern mining-BAseD Community discoverer in mUltidimensional networkS (ABACUS for short) for partition integration. ABACUS is able to extract communities from MRNs based on frequent closed itemsets mining from single-relational community memberships.

The key to understand ABACUS is how to use transaction model to represent the results of community partition on each slice. The transaction model builds the bridge from partition integration to frequent itemsets mining. In ABACUS, each transaction, i.e., a record in the transaction database, corresponds to a node, where an item is a pair (dimension, community), expressing the membership of the node in the various dimensions. In the field of association analysis, frequent closed itemsets provide a minimal representation of itemsets without losing their support information. An itemset $X$ is closed if none of its immediate supersets has exactly the same support count as $X$ [25]. Therefore, under the transaction model adopted by ABACUS, a frequent closed itemset consists of a set of nodes, and it represents nodes in this itemset are frequently grouped together in different slices. As the support count of an itemset exceeds a pre-defined threshold, nodes in this itemset are extracted as a community.

Consider a simple MRN with six nodes and three relations. After the community partition of each slice is ready, the lattice view of pattern mining in ABACUS is shown in Fig. 4.4. “TID” corresponds to the node ID, and each item corresponds to a community. For instance, “A=VLDB-1” represents the #1 community in “VLDB” relation. The mined frequent closed itemsets shown in dotted-lined rectangles are communities extracted by ABACUS. For example, nodes 1, 2 and 3 share their memberships to the communities “VLDB-2” and “KDD-1”, which implies they are closely interrelated.

Although the pattern mining method is novel and enlightening, it leaves us too many research issues. On the one hand, a large number of studies pointed out that the frequent patterns [8, 25, 33], i.e., itemsets, are not always interesting, such as the
famous “coffee–tea” example. Also, interesting patterns might always infrequent. A set of well-accepted criteria were established for evaluating the interestingness of patterns, such as lift, cosine, All-confidence, etc., but as a pattern describes a set of nodes in social networks, how to measure the interestingness from the community perspective, and thus how to mine them in an efficient manner deserve the deep research. On the other hand, patterns are always nested and most of them may be very small. So, how to exploit patterns to discover communities with rationale size, or even to reveal hierarchical community structures also deserved the future research.

### 4.5.2 Consensus Clustering Based Method

Consensus clustering, also known as cluster ensemble or clustering aggregation, aims to find a single partitioning of data from multiple existing basic partitionings [10, 14, 31]. In the literature, lots of consensus clustering algorithms have been proposed, such as the graph-based methods [23], the co-association matrix-based methods [10], the K-means-based methods [31], and other heuristic approaches [9]. However, most research on consensus clustering focused on text data, and few attentions have been paid to exploiting consensus clustering for community discovery on graph data.

Recently, Lancichinetti and Fortunato [15] proposed the Algorithm which Integrates Consensus Clustering in a given community detection Method (AICCM for short) on single-relational networks. Consensus clustering is used by AICCM to enhance the quality and robustness of any given community detection method. We introduce the main idea of AICCM by an example as shown in Fig. 4.5. First, it employed a given community detection method for several times to obtain several basic partitionings, e.g., (1)–(4) in the left part of Fig. 4.5. Second, it assembled all
basic partitionings to get an overall similarity matrix $S$. Let $H^d \in \{0, 1\}^{n \times k^d}$, $1 \leq d \leq m$ denote a binary membership indicator matrix of a basic partitioning. $S$ is computed as

$$S = \frac{1}{m} HH^T = \frac{1}{m} \sum_{d=1}^{m} H^d (H^d)^T.$$  \hspace{1cm} (4.26)

The similarity matrix $S$ can be regraded as an adjacent matrix of a weighted consensus graph illustrated on the right part of Fig. 4.5, where the thickness of each edge is proportional to its weight. In the consensus graph the cluster structure of the original network is more visible: the two communities have become “weak” cliques, with “heavy” edges, whereas the connections between them are quite weak. Interestingly, this improvement has been achieved despite the absence of two inaccurate partitions $H^3$ and $H^4$. After that, AICCM again applies the same algorithm on this consensus graph four times to acquire four partitions. If the partitions are all equal, stop; otherwise, repeat the above steps.

Applying consensus clustering for partition integration on MRNs is straightforward, when we take the basic partition $H^d$ as the results of community detection on $d$th relation. In [26], three graph-based methods [23], including Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph-Partitioning Algorithm (HGRA), and Meta-CLustering Algorithm (MCLA), have been used for partition integration.
CSPA first assemble basic partitionings as a consensus graph being same as AICCM, and then apply any community detection method on that graph. HGPA is to repartition the data using the given clusters as indications of strong bonds, and to formulate consensus clustering as the hypergraph partitioning problem. Analogously, MCLA is to group and collapse related hyperedges, and thus to assign each object to the collapsed hyperedge in which it participates most strongly. The hyperedges that are considered related for the purpose of collapsing are determined by a graph-based clustering of hyperedges.

Despite of much above-mentioned research efforts, some research issues still exist. The computed similarity matrix by CSPA is usually much denser, which would make the application of any clustering algorithm computationally expensive. HGPA may be not appropriate if the natural MRN data clusters are highly imbalanced. Moreover, MCLA is more data-dependent, i.e., performing poorly on some benchmark datasets [29]. Meanwhile, the efficiency is always a major concern of consensus clustering, since it is a combinatorial optimization problem in essence. Therefore, when facing graph data especially big graph data, how to design both effective and efficient consensus clustering algorithms for partition integration remains a great challenge.

4.6 Experimental Networks

Over the past few decades, a large amount of real-world single-relational networks have been published, containing from hundreds of nodes to millions of nodes. A typical network repository is Stanford large network dataset collection (http://snap.stanford.edu/data/). However, the public real-world datasets for MRNs are absolutely rare. To the best of our knowledge, only one dataset named 3T [17] containing multiple social relationships was collected directly from people’s daily life, but unfortunately, it is now not public to researchers due to privacy concerns. There are commonly two ways for researchers to construct MRNs to validate their new methods or algorithms: using the synthetic data, and constructing MRNs based on attribute values collected from the real-world websites [4, 6, 7, 20, 28, 32].

The synthetic data is usually simple and relatively smaller, but the ground-truth is known, which enables researchers to utilize it for basic performance comparison. Otherwise, MRNs constructed on real-world attributes are often complex and large, but the ground-truth is unknown. Thus, the internal measures and semantic information are often used for performance validation. The DBLP is the most used dataset for constructing the MRN. Some MRNs can be directly crawled from websites, such as YouTube used in [28]. In this section, we introduce the technique for constructing the MRN on DBLP data, which can also be applied to other similar data. Then, we discuss how to construct MRNs based on forum data.
4.6.1 Constructing MRN on DBLP Data

The DBLP becomes the most popular data for performance validation in MRN research field due to the following reasons. First, the semantic of authors is clear. For example, Jiawei Han is a well-accepted authority in data mining. Second, the categories associated with conferences and papers are easier to be obtained. One typical source for rankings and categories of authoritative conferences in computer science is the website (http://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html).

Publication information on DBLP includes title, authors, reference list (i.e., citations), conference/journal name, abstract, and classification categories. Based upon these attributes, many kinds of different MRNs can be constructed. In community discovery, nodes often correspond to authors, and multiple relations often correspond to categories such as data mining, databases, multimedia, and so on. Thus, the attributes can be used usually include the publication number of every author on every category, or citations of an author through a category. So, the key task of constructing a MRN is how to compute the weight between author in each category, i.e., relation. Let $p_{i,d}$ and $p_{j,d}$ denote the attribute values of two researchers $i$ and $j$ on $d$th category. Tensor $A$ is computed as \[ A_{i,j}^d = e^{-(p_{i,d} - p_{j,d})^2}. \] (4.27)

A variant of Eq. (4.27) is presented in [32].

\[ A_{i,j}^d = e^{-(2(p_{i,d} - p_{j,d})/p_{i,d} + p_{j,d})^2}. \] (4.28)

Although using the difference value to measure the relation strength in MRN is widely-used, it may not reflect the truth in some cases. For instance, node $i$ is an famous expert who has published a great many papers in a category, and node $j$ is $i$’s student who published few papers. Thus, according to Eq. (4.27) or (4.28), the relation strength between $i$ and $j$ is weak, but they are strongly connected in the real world. So, how to define the relation strength (or similarity) based on attributes remains an important issue to be solved. Despite of this, it is interesting to use this technique to transform multivariate UCI datasets with ground-truth to MRNs. For example, the Iris dataset has 4 attributes and 150 instances, and it is used to construct a MRN with 150 nodes and 4 relations by computing the relation strength between instances on every relations.

4.6.2 Constructing MRN on Forum Data

Online forums (e.g., Google Groups, HardForum, and Tianya) are appealing places for members of which to communicate due to their openness and freedom.
Generally, forum data include two categories: the initial article and the reply article. The initial article is the initiator/organizer of a topic, while a reply article comments on an initial article or another reply article so as to continue the discussion. Therefore, given an initial article (i.e., a topic), lots of forum users will be involved into discussion (i.e., interactions) under that topic. These interactions are indicated by the replier ID, recipient ID, reply time, and the content contained in each reply article. Mining text data about discussion articles can reveal multiple hidden interactions between users, which implicitly forms a MRN.

We have conducted some initial research on forum data processing (e.g., Tianya). In particular, we presented several kinds of measurement models for various interactions between users, including undirected dense/sparse (UDN/USN), interest(IN), emotion(EN) and similar view networks(SVN) [5, 35], which can be used to construct a MRN on forum data. According to community discovery and evaluation on each dimension, we obtained some interesting findings. First, though community structures are not very clear on interest and similar view dimensions, users in the same group tend to have similar interests or consistent perspectives. Second, the emotional dimension played the most important role among multiple relations, on which the identified communities are strongly segregated with each other. This implies the emotional information on forum can better guide the community discovery.

Here, we introduce a possible way for constructing MRNs using the data from Tianya, a popular bulletin-board service in China. Let each node in the MRN \( i \in V \) corresponds to a registered user ID on Tianya forum, and each edge \((i, j) \in E\) represents a specific interaction mined from users’ comment activities. Also, Let \( n_{i,j} \) be the number of times that user \( i \) writes a comment to user \( j \).

We construct a five-relational network using the data from Tianya, a popular bulletin-board service in China. Every registered user identification (ID) in Tianya forum corresponds to a node \( i \in V \) in a MRN. Edges \((i, j) \subseteq E\) represent some social relations between two users that results Tensor \( A \) is computed as

\[
A^d_{i,j} = \begin{cases} 
  n_{i,j} + n_{j,i} & \text{if } (i, j) \in E_{UDN} \\
  \min(n_{i,j}, n_{j,i}) & \text{if } (i, j) \in E_{USN} \\
  \sum_{p \subseteq P_{i,j}} \min(n_{i,p}, n_{j,p}) & \text{if } (i, j) \in E_{IN} \\
  \text{trust}_{i,j} & \text{if } (i, j) \in E_{EN} \\
  \text{AC}_{i,j} & \text{if } (i, j) \in E_{SVN}
\end{cases}
\]  

(4.29)

where \( P_{i,j} \) is the set of users to whom user \( i \) and \( j \) together comment. The sign of a given comment can be defined as positive or negative based on the average semantic orientation of seed emotional words in the review [30]. If we use \( E_{i,j,k} \) to represent the emotion value of \( k \)th reply from user \( i \) to \( j \), the “trust” between user \( i \) and user \( j \), \( \text{trust}_{i,j} \), can be defined as

\[
\text{trust}_{i,j} = \frac{\sum_{k=1}^{n_{i,j}} E_{i,j,k} + \sum_{k=1}^{n_{j,i}} E_{j,i,k}}{n_{i,j} + n_{j,i}},
\]  

(4.30)
Bu et al. [5] further indicated that the similar user pair should have similar interests and consistent perspectives to most topics they participate together. So, the attitude consistency of user $i$ and $j$, $AC_{i,j}$ is defined as:

$$AC_{i,j} = \frac{\sum_{t \subseteq T_{i,j}} \sum_{p \subseteq P_{i,j}^t} \sigma(\xi_{i,p}^t, \xi_{j,p}^t)}{\sum_{t \subseteq T_{i,j}} \text{card}(P_{i,j}^t)},$$

(4.31)

where $\xi_{i,p}^t$ is the perspective from user $i$ to user $p$ under the topic $t$. $P_{i,j}^t$ is a user set, it includes the users to whom user $i$ and $j$ together comment in the discussion of topic $t$. $T_{i,j}$ is a topic set, it includes the topics which are together discussed by user $i$ and $j$. $\text{card}(\cdot)$ returns the total number of users in the user set. And $\sigma(x, y)$ is a judgment function determined by $x$ and $y$, which obeys:

$$\sigma(x, y) = \begin{cases} 1 & \text{if } x > 0.5, \ y > 0.5 \text{ or } x < 0.5, \ y < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

(4.32)

The range of $AC_{i,j}$ is between 0 and 1, and a higher value corresponds to a greater degree of attitude consistency of the given user pair to their together-reply topics/users.

Figure 4.6a shows a tree structure corresponding to a small thread of depth 4. Labels denote the user who writes the contribution and valid comments are shown within the gray region. The post triggers three responses from users $A$, $C$ and $D$.

![Diagram](https://via.placeholder.com/150)

**Fig. 4.6** An illustration example for the Tianya MRN. a An example of discussion list. b Undirected dense network. c Undirected sparse network. d Interest network. e Emotion network. f Similar view network.
At the second nesting level, eight comments appear. At the third level, there are still seven comments and finally, there is one last comment from C. The attitude of every comment can be represented using + or −, with + denoting a user is supportive to the viewpoint and − otherwise. The corresponding undirected dense/sparse, interest, emotion and similar view networks excavated from original thread of comments are shown in Fig. 4.6b–f respectively. The weight attached to each edge represents the strength of connections between the corresponding members.

4.7 Summary

This chapter defined the community discovery problem on multi-relational networks and reviewed representative research on this problem. We start by introducing the generalized modularity of the MRN, which paves the way for applying modularity optimization-based community detection methods on MRNs. According to the classification of the mainstream method—integration methods, including network integration, utility integration, feature integration, and partition integration. We introduced several co-ranking frameworks on MRNs, which could learn the biased weights of every relations for more precise integrations. We then discussed two typical methods for the partition integration, including frequent itemsets mining based method and consensus clustering based method. Last but not the least, due to the lack of real-world MRNs, we presented several techniques for constructing the MRN based on both multivariate data and forum data. This provides operational and practical experimental techniques to MRN-related research.

Acknowledgments This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grants 71571093, 71372188 and 61502222, National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035, National Key Technologies R&D Program of China under Grant 2013BAH16F01, National Soft Science Research Program under Grant 2013GXS4B081, the Program of Natural Science Foundation of Zhejiang Province under Grant LY13F020008, the Ministry of Education of Humanities and Social Sciences Project under Grant 14YJCZH235, Industry Projects in Jiangsu S&T Pillar Program under Grants BE2014141 and BE2012185, and Key/Surface Project of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grants 12KJA520001, 14KJA520001, 15KJB520012 and 14KJB520015.

References