

Chapter 2

Community Detection in Bipartite Networks: Algorithms and Case studies

Taher Alzahrani and K.J. Horadam

Abstract There is increasing motivation to study bipartite complex networks as a separate category and, in particular, to investigate their community structure. We outline recent work in the area and focus on two high-performing algorithms for unipartite networks, the modularity-based *Louvain* and the flow-based *Infomap*. We survey modifications of modularity-based algorithms to adapt them to the bipartite case. As Infomap cannot be applied to bipartite networks for theoretical reasons, our solution is to work with the primary projected network. We apply both algorithms to four projected networks of increasing size and complexity. Our results support the conclusion that the clusters found by Infomap are meaningful and better represent ground truth in the bipartite network than those found by Louvain.

2.1 Introduction

A very large number of clustering algorithms is available for community detection in networks. These algorithms try to identify subgraphs (often called communities, clusters or modules) which are more tightly connected internally, according to a particular measurable rule, than they are connected to the rest of the network. The practical aim is to derive a coarse-grain picture of a real large-scale network which will aid understanding of its hierarchical structure. However, there may not be a strong correlation between the clusters found by an algorithm and the ground truth of hierarchical structure within the network, since real-world community formation may be a result of many interacting and potentially unmeasurable rules. In any case, the ground truth in a real network may not be directly discernable by virtue of the network's size and complexity. Thus we would like to have some confidence in the meaningfulness of the optimal partition arrived at by a clustering algorithm.

T. Alzahrani · K.J. Horadam (✉)
School of Mathematical and Geospatial Sciences, RMIT University, Melbourne,
VIC 3001, Australia
e-mail: kathy.horadam@rmit.edu.au

T. Alzahrani
e-mail: taher.alzahrani@rmit.edu.au

This issue becomes more complicated for bipartite (or, more generally, multipartite) networks. Simple, or unipartite, networks are the typical framework for complex network study. However, many complex networks can best be described as bipartite [36]. In a bipartite network, the node set consists of two disjoint sets of nodes such that links between nodes may occur only if the nodes belong to different sets. Examples of bipartite networks are citation networks between authors and published papers in academia, recommendation systems in online purchasing, protein interaction networks in biological science and movie-actor networks in social networks.

Obviously, every bipartite network can be treated as unipartite by ignoring the node partition, but in the last few years, there has been increasing motivation to analyse bipartite networks as a separate network category, and in particular to investigate their community structure.

Usually one set of nodes in a bipartite network, the *primary set* P , is of more interest for a particular purpose than the other, the *secondary set* S . In this case, P may be treated as the node set of a unipartite *projection* network, whose edges are derived from linking information in the bipartite network. A battery of unipartite clustering algorithms may then be applied directly to the projection. The rôles of the two node sets can be switched for different applications.

Many real networks arise naturally as projections of bipartite networks. It has also been argued [20] that *every* complex network is a projection of a bipartite network constructed from its node set (as P) and from a set of cliques that it contains (as S), and that this bipartite model explains many of the network's main properties.

There are different ways of defining the edges in a projection on P . Furthermore, the structure of the projection on P will depend on S in important ways: in [32] it is shown that the degree distribution of the projection on P depends very strongly on the degree distribution of S .

So there are really two approaches to identifying clusters in a bipartite network: the first, and more common, is when our real interest is in community structure within the primary node set P ; and the second is when our real interest is in bipartite communities within the whole network.

In this chapter we focus on the first approach. We outline recent work on community detection algorithms for unipartite networks and how they have been adapted, or else cannot be applied, to bipartite networks. We apply the two highest-performing algorithms to several projected networks. Our results support the conclusion that the clusters found by the flow-based algorithm *Infomap* better represent ground truth in the bipartite network than those found by the best modularity-based algorithm *Louvain*.

The chapter is organised as follows. In the next Sect. 2.2 we give an overview of clustering algorithms which are used for partitioning nodes into non-overlapping communities in a large and complex unipartite network. For unipartite networks, two approaches to community detection have been very popular, one based on modelling the clustering structure and one based on extracting it from flow calculations on the network. The best algorithms to cluster very large networks using each approach, compared using the LFR benchmark datasets [27], are now referred to as the Louvain algorithm [9] and the Infomap algorithm [47].

Section 2.3 surveys how these and other algorithms have been modified for the important class of bipartite networks. As Infomap cannot be directly applied to bipartite networks, in Sect. 2.4 we describe its application to the network found by weighted projection onto the primary node set P . In Sect. 2.5 we look further at the critical issue of whether the clusters found by Infomap in the weighted projection network make sense: that is, whether or not they represent some sort of ground truth. We present several case studies to support the proposition that they do. Finally, in Sect. 2.6 we describe our intended solution to adapt Infomap to bipartite networks and propose a list of further research questions.

2.2 Community Detection Algorithms

In complex networks a community (or cluster, or module) is a fundamental qualitative concept for which there is still no single accepted definition. It may be a node based idea, as we use here, or an edge based one.

In a node based definition, a cluster is a set of nodes which connect more to each other than to other nodes of the network, based on the idea that they share the same resources or have similar properties. This kind of definition is widely accepted and used. A well-known quality function that evaluates clusters based on this idea is modularity [35].

On the other hand, in an edge based definition, a cluster is a group of edges rather than of nodes [1, 13]. The classification of edges into groups is based on their similarity through sharing nodes of the network. This definition is useful in dealing with overlapping communities, where each node inherits membership from all its incident edges and can belong to multiple communities according to the similarity between these edges.

Such different definitions lead to rapid evolution of a vast number of cluster detection techniques [15]. From our point of view, the choice of definition depends on the context and application requirements for a particular network. For example, on the World Wide Web (WWW) a cluster can be looked at as information or as physical links and routers connecting to each other. Scientific collaborations can be classified as clusters of scientists, clusters of papers or both. Social network clusters can be defined as people relating to each other or as interests that are shared by a group of people.

In this section we will first establish some basic concepts about clusters and comparison of partitions, then describe the LFR benchmarks for testing performance of community detection algorithms. We follow with a description of modularity-based algorithms and the problem of the resolution limit and then conclude by outlining flow-based algorithms.

We will use the following notation throughout: in a network $G = (V, E)$ with node set V and edge set $E \subseteq V \times V$, we set $n = |V|$, $m = |E|$ and let $A = [A_{ij}]$ represent the network's adjacency matrix. (If the network has multiple edges then

A_{ij} is the number of edges from node i to node j .) Node i will have degree k_i , and we observe that $k_i = \sum_j A_{ij}$. The complete graph or clique on n nodes is denoted K_n .

In a bipartite network G , $V = P \vee S$ and $E \subseteq P \times S$ is the set of edges.

2.2.1 Comparing Clusters and Partitions

Clusters of nodes can be regarded as strong or weak. Probably the simplest definition of a strong cluster is a set of nodes which forms a clique, that is, the subgraph they induce is complete [39]. However there are less absolute ideas of community which are used more commonly.

The crucial idea behind strong and weak clusters in [43] is the degree k_i of a node i that belongs to the cluster. For a particular cluster c to which node i belongs, we separate k_i into two parts: the number of edges $k_i^{in} = \sum_{j \in c} A_{ij}$ connecting node i to other nodes in c , and the number of edges $k_i^{out} = \sum_{j \notin c} A_{ij}$ connecting node i to the nodes in the rest of the network. A strong cluster has to satisfy the condition:

$$k_i^{in} > k_i^{out}, \quad \forall i \in c \quad (2.1)$$

that is for each $i \in c$, it must have more edges to the nodes within the cluster c than edges connecting to the rest of the network. A weak cluster is defined as:

$$\sum_{i \in c} k_i^{in} > \sum_{i \in c} k_i^{out} \quad (2.2)$$

that is, the sum of all degrees for all nodes within c is larger than the sum of all degrees outgoing to the rest of the network. Obviously, a strong cluster is a weak cluster as well, but the converse is not true.

An alternative definition is proposed in [24], which relates the cluster under consideration to each other cluster and not to the whole network. Here, a strong cluster is a set of nodes where each node's degree within the cluster must be at least as large as its degree toward any other cluster in the network:

$$\forall i \in c, \quad k_i^{in} \geq \max_{c' \neq c} \left\{ \sum_{j \in c'} A_{ij} \right\}. \quad (2.3)$$

A weak cluster accordingly is one where the sum of all degrees within the cluster should be at least as large as the sum of degrees outgoing to each cluster in the network:

$$\sum_{i \in c} k_i^{in} \geq \max_{c' \neq c} \left\{ \sum_{i \in c} \sum_{j \in c'} A_{ij} \right\}. \quad (2.4)$$

Another approach, used in [18], defines a strong cluster using the betweenness centrality measurement. The betweenness is calculated for a given edge e as the number of shortest paths between every pair of nodes in the network that run through e . By iteratively removing the edges with highest betweenness centrality, components of the network will split from each other forming clusters. These hierarchies of clusters are represented in a binary tree, with nodes in a cluster more closely connected compared with other nodes in the network.

The aim of clustering algorithms is to reveal the topological structure of the network. To evaluate the communities detected by these algorithms, similarity measures have been proposed in order to assess the fit of the partition found with a desired one. A similarity measure very widely used for this purpose is Normalized Mutual Information (NMI), which tests the “goodness” of a detected partition by measuring the common information it shares with a targeted partition. The version of NMI that has been widely accepted in the literature, particularly in the LFR benchmark [27], is from [11]: given two partitions C and C' ,

$$I_{norm}(C : C') = \frac{H(C) + H(C') - H(C, C')}{(H(C) + H(C'))/2} \quad (2.5)$$

where $H(C) = -\sum_c P(c) \log P(c)$ is the Shannon entropy for partition C and $H(C, C')$ is the joint entropy between the two partitions. I_{norm} equals 1 if the two partitions are identical and 0 if they independent.

Another approach uses the Jaccard similarity coefficient for comparing two partitions of the network [8]. It is defined as the ratio of the number of node pairs classified in the same cluster in both partitions, over the number of node pairs which are classified in the same cluster in at least one partition. Let us say that a_{11} is the number of node pairs which are in the same cluster in both C and C' , a_{10} indicates the number of node pairs that are put in the same cluster in C but not in C' and a_{01} is the number of node pairs put in the same cluster in C' but not in C . The Jaccard similarity coefficient for C and C' is:

$$J(C, C') = \frac{a_{11}}{a_{11} + a_{01} + a_{10}} \quad (2.6)$$

The ratio of the Jaccard similarity coefficient for the two partitions C and C' is between 0 and 1. When $J(C, C') = 1$ the clusters in C are identical to the clusters in C' , while $J(C, C') = 0$ indicates independent clusters in both partitions, with no overlap at all.

Of course the ideal situation for measuring performance of a community detection algorithm is based on the ground truth. It requires deep knowledge of the formation of relations within and between clusters. Although it is excessively time consuming, and impractical or impossible in large networks, the result is much more accurate and more meaningful. In this chapter, we follow this approach as it provides significant outcomes.

2.2.2 Benchmarks and Performance

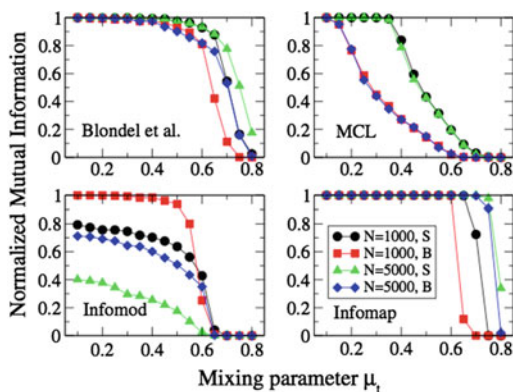
The LFR benchmark [27] allows authors of community detection algorithms to test their algorithms and evaluate the communities they have detected. It relies on creating an artificial network belonging to a “planted ℓ -partition” model, which generates a network with a given community structure. The node partitions generated in such a network can have different sizes, and different nodes can have different degrees.

This benchmark suite has the ability to test large networks of 10^3 to 10^5 nodes, to deal with overlapping communities and with both directed and undirected networks. Its novelty is that it is possible for both node degree distribution and community size to follow the power law distribution.

To run a test on the LFR benchmark, the mixing parameter μ has to be tuned at different values in the range $[0, 1]$. This mixing parameter μ is the ratio of the number of external neighbors (k^{out}) of each node by its total degree. A small value of μ indicates well separated communities, however with a high value of μ , communities overlap more and more and the community structure is weaker. This mixing parameter allows the strength of the community structure to be controlled, to be compatible with realistic topological properties. The test algorithm is run against LFR networks constructed using a selection of values for μ and the partition it finds is compared with the planted partition using NMI. The complexity of the LFR benchmark is linear in the number of edges of the constructed network, which makes performing such testing fast enough to analyse and study.

A comparative analysis of the performance of 12 community detection algorithms appears in Lancichinetti et al. [28, 29]. This study enables us to compare the stability and the accuracy of algorithms by testing them against heterogeneous distributions of node degree and community size. The outcome of this study is that the Infomap algorithm is the best algorithm to cluster very large networks, followed by the Louvain algorithm (but see the next subsection) and a Potts model algorithm. Figure 2.1 from

Fig. 2.1 The performance of four algorithms against LFR benchmark partitions. Infomap is at *bottom right* and Louvain (Blondel et al.) at *top left* [28]



[28]¹ shows the comparative performance for various μ . We describe the first two algorithms in the following subsections.

Furthermore, a recent evaluation for 11 algorithms appears in [38] where the emphasis is on the strength of community structure. It used the artificial networks generated by the LFR benchmark, where node degrees and community sizes are both power-law distributed, with a different mixing coefficient, and again the NMI is used to assess the performance of the algorithms. This evaluation concludes that Infomap is the leading algorithm on performance among all 11 algorithms.

2.2.3 Modularity Based Algorithms and the Resolution Limit

Girvan and Newman [35] initiated recent work on detecting and evaluating communities in large networks. They introduced a fast greedy technique which relies on maximising a quality function called *modularity*, defined for a partition C as

$$Q(C) = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c(i), c(j)) \quad (2.7)$$

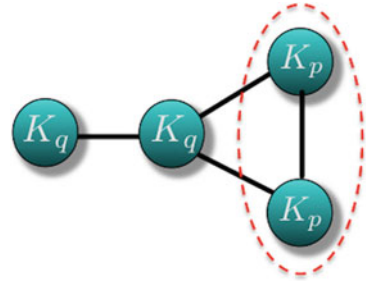
where $c(i)$ is the community to which node i is assigned, and the Kronecker delta function $\delta(c(i), c(j)) = 1$ if nodes i and j belong to the same community and 0 otherwise. The complexity of the Girvan-Newman algorithm is $O(n^3)$ and it is limited to networks with around $n = 10^3$ nodes.

Many efforts have been devoted to upgrade the computational time of modularity optimization, and extend the limit of network size that can be clustered. For instance, the Radicchi et al. [45] algorithm, in the spirit of Girvan-Newman, iteratively removes edges, but in this case removes the edges with highest clustering coefficient instead of edges with highest betweenness. The complexity of this algorithm is $O(n^2)$ which is an improvement on the greedy technique. Another example of an algorithm that takes modularity optimization as its main quality function is that of Guimera and Amaral [21].

The Walktrap algorithm proposed by Pons and Latapy [41] uses random walks to define a distance which measures the structural similarity between vertices and between communities. It is based on the idea that at some stage a random walker tends to be trapped in dense part of a network corresponding to a community. Starting from an initial assignment of each node to its own community, communities are merged according to the minimum of their distances and the process iterated. The bottom-up hierarchy is represented in a dendrogram and the algorithm stops when a partition with maximum modularity is obtained.

¹Figure reprinted with permission from Ref. [28]. ©2014 by the American Physical Society.

Fig. 2.2 Maximisation of modularity Q will fail to identify cliques in this example, e.g. if $q \gg p$, there is higher modularity for the pair of cliques K_p joined by a single edge than for the cliques themselves



However, modularity optimisation algorithms are subject to a resolution limit in the size of communities they can detect. Fortunato and Barthelemy [16] showed that communities with internal edge numbers $\leq O(\sqrt{m})$ may not be detected. Small strong communities in large networks may fail to be resolved, even when they are well defined. An illustrative example appears in Fig. 2.2. This is a definite drawback for modularity-based algorithms.

The fast modularity optimization algorithm by Blondel et al. [9], known as the *Louvain* algorithm, has one of the best results in the comparison tests [28]. The first phase of this algorithm starts by assigning each node in the network to its own community, then merging neighboring nodes that maximise value in the modularity equation. The second phase starts by dealing with previously found communities as super-nodes in a new network and repeats the first phase on this new network by merging two super nodes to achieve a higher modularity value. These steps are repeated iteratively until the maximum modularity is reached, resulting in multi-levels of communities, as super-nodes. The complexity of the Louvain algorithm is linear in the number of edges in the network, that is $O(m)$. The authors claimed the multi-level nature seems to circumvent the resolution limit problem of modularity and this appeared to be born out by its high performance evident in Fig. 2.1.

However, a very recent acknowledgement by Lancichinetti et al. [29] admits that in Fig. 2.1 they did not use the subsequent iterates of the Louvain algorithm in determining its performance, only the first phase, because the performance of the final level would be very poor, owing to the resolution limit.

2.2.4 Minimum Description Length Based Algorithms

The stochastic block model of Peixoto [40] employs minimum description length (MDL) to describe the structure of a network, through compressing the total amount of information on the network. It identifies the blocks (communities) for a network without needing to specify the number K of blocks in advance. However, there is a resolution limit in detecting the blocks which is similar to the resolution limit in modularity based algorithms: the maximum detectable block number K scales as \sqrt{n} for a fixed average degree.

The map equation method proposed by Rosvall and Bergstrom [47], known as *Infomap*, identifies communities according to information flow in the networks. Infomap has two main steps, a deterministic greedy search algorithm and then a simulated annealing approach to refine the results obtained. In its greedy search step the algorithm starts by calculating the ergodic node visit frequencies using a transition matrix to create the stationary distribution for the network.

This approach uses Huffman codes [23] to give short codewords for commonly visited nodes, and long codewords for rarely visited nodes. The quality function used to evaluate a partition is again the minimum description length MDL [19]. It measures the average length $L(C)$ in bits per step of a random walk on the network with a node partition $C = \{c_1, \dots, c_l\}$.

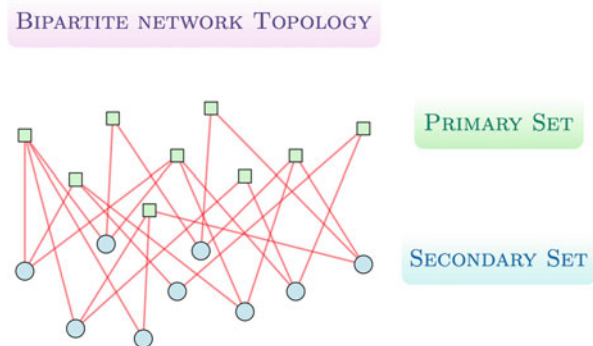
$$L(C) = q_{\sim} H(C) + \sum_{i=1}^l p_{\odot}^i H(P^i) \quad (2.8)$$

This equation has two parts: the first one is to explain the movements between the communities, where q_{\sim} is the probability that a random walker switches communities and $H(C)$ is the entropy of the community index codewords. The second part explains movements within the communities, where p_{\odot}^i is the fraction of the movements within community c_i and $H(P^i)$ is the entropy of the movements within community c_i . The complexity of the Infomap algorithm is $O(m)$.

2.3 Algorithms for Bipartite Networks

In this section we discuss community detection algorithms that are intended for bipartite networks, and the fact that the best-performed algorithm, Infomap, cannot be applied to them. Figure 2.3 illustrates the structure of a bipartite network.

Fig. 2.3 Bipartite network structure



2.3.1 Modularity-Based Algorithms

Most authors follow the modularity method of Newman and Girvan [35] to find communities in bipartite networks. Since bipartite networks have two distinct node sets and edges only connect nodes from different sets, modularity optimization needs to be modified to identify communities in this kind of network. Guimera et al. [22] introduced a modularity measurement for bipartite networks and checked its performance against the communities in the weighted projection on P detected directly by modularity maximisation. They found no difference between these and the communities in P that resulted after projecting the communities they found in the bipartite network.

In [6], Barber developed the modularity matrix for bipartite networks, inspired by Newman's idea of a modularity matrix [34]. The modularity equation from Newman [34] takes the following form (cf. (2.7)):

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(c(i), c(j)) \quad (2.9)$$

where P_{ij} is the probability of an edge existing between i and j . Barber claims that there is a profound impact on the modularity using a normal null model in this equation, since it assigns edges at random with the expected degree of model vertices constrained to match the degrees in the actual network. Thus, he defines a null model that obeys the requirement of bipartite networks.

2.3.2 Label Propagation Algorithms

A different technique for detecting communities in unipartite networks is the Label Propagation Algorithm (LPA), proposed by Raghavan et al. [44], which uses the local network structure as a guide for finding communities in unipartite networks. LPA doesn't perform as well as Louvain and Infomap on the LFR benchmark [27]. Barber and Clark [7] introduced an extended version of LPA, denoted LPAb, for bipartite networks.

The LPAb is very fast on a bipartite network and it is an efficient method of detecting communities through maximizing the modularity optimization. Initially, it starts by assigning a unique label for each node in bipartite network that reflects their node set, so we have two different colors, say nodes in P are labelled red and in S are labelled blue. Then, nodes update their label at each step in random sequences to obtain maximization in bipartite modularity. These processes are repeated iteratively until a local maximum of bipartite modularity is reached. At the end, modules are identified as a group of nodes having the same labels. The speed of LPAb makes it the "fastest bipartite modularity optimization algorithm" [11] because the computational

time for LPAb is $O(m)$. Liu and Murata introduce an improved version of LPAb, called LPAb+ [31], which is claimed as the most reliable algorithm with highest bipartite modularity.

2.3.3 Statistical Modelling and MDL Algorithms

The MDL based stochastic block model of Peixoto [40] can be applied to a bipartite network without specifying that it is bipartite. When applied to the IMDB bipartite network (discussed in Sect. 2.5.3 below) the resulting communities from this block model fully reflect the bipartite nature, as the detectable communities partition P and S separately.

A statistical modelling approach to community detection in bipartite graphs has been proposed in [3]. The paper first surveys the statistical models used for modelling networks where actors attend events (some of these models are not intended for community detection), and of which only one (the exponential random graph or p^* model) had previously been applied to the benchmark Southern women network discussed in Sect. 2.4.2 below. It discusses a latent class model, which is a “mixed Rasch model” where the number of communities, K , is an initial (unknown) variable, and particular choices of K are fitted by assigning different event attendance probabilities among groups, but identical attendance probabilities within groups. An assumption of the model is that actors attend events independently. The choice of K is discussed at some length.

2.3.4 Infomap and the Convergence Problem

Infomap, the best performing algorithm for community detection, cannot run as intended on a bipartite network. The stationary distribution (probability of being at node i) for random walks on any network is given by the probability [30]:

$$\pi_i = \frac{k_i}{2m}, \quad i = 1, \dots, n. \quad (2.10)$$

On a bipartite network, a walk alternates between the two node sets, so, while the stationary distribution is computable, the walk doesn’t converge to it as time tends to ∞ independent of the start node. For example, if the random walk starts in one node set of a bipartite network, then it will always be in that set after an even number of steps, so the probability of being at a node in that set is zero at odd time steps. Infomap fails at its first step on a bipartite network. Thus, we can not implement Infomap on bipartite networks because of periodicity.

2.4 The Weighted Projection Approach

We cannot apply Infomap to bipartite networks directly but we can certainly apply it to a (weighted) projection onto P . Guimera et al. [22] found no difference in the node communities detected in P whether they resulted from modularity maximisation after projection, or projection after bipartite modularity maximisation. The projection method has been used for a long time in recommendation systems in the business area. Its strength is the idea that the emphasis is usually on one of the two node sets. These sets can be switched for different applications. So a weighted projection method allows us to investigate bipartite networks using powerful one mode algorithms, after a transforming process.

A projection of P in $G = (P \vee S, E)$ is a graph $G_P = (P, E_P)$ in which two nodes i and $j \in P$ are linked together if they have at least one neighbor in common in S . A projection can be weighted or unweighted but weighted projections are usually regarded as more representative of the link information in the bipartite network. Two nodes in P are more likely to have a meaningful link in reality if they have many neighbors in common, and this information should not be lost. The number of common neighbours can be represented by multiple edges between the nodes, or else by a weighted single edge between the nodes. Moreover, the information that a node in P connects to a node of degree 1 in S should not be lost.

In this section we describe the weighted projection algorithm we use, and we compare its community detection outcomes with others in the literature on a small database, which is nonetheless a benchmark for bipartite clustering techniques, the ‘‘Southern women’’ database [12].

2.4.1 Description and Method

Multiple edges are computationally time-consuming to process, and here we use weighted edges. Moreover, Infomap and Louvain can accept weighted networks as input.

Given G , the adjacency matrix for G_P is defined by:

$$A_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ have a common neighbor} \\ 1, & \text{if node } i \text{ has a neighbor which has no other} \\ & \text{neighbors in } P \text{ (resulting in a self loop at } i \text{)} \\ 0, & \text{otherwise} \end{cases}$$

For node $i \in P$, let $\Gamma(i)$ denote the set of neighbors of i ; all these are nodes in S . To measure similarity between distinct nodes i and j in P we choose the common neighbors index,

$$W_{ij} = |\Gamma(i) \cap \Gamma(j)|, \quad i \neq j \quad (2.11)$$

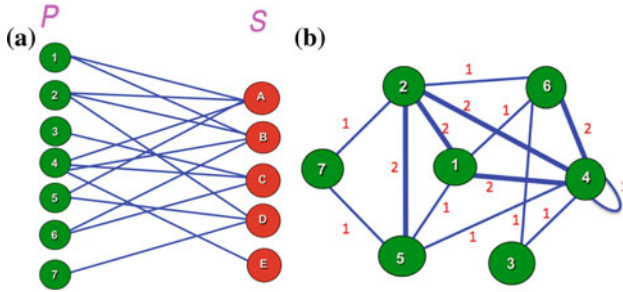


Fig. 2.4 Example of weighted projection. **a** Bipartite network with $n = 12$, $m = 15$, $|P| = 7$ and **b** weighted projection of P using (2.11) with $|E_P| = 19$

due to its simplicity and efficiency [48] on large scale networks. Then W_{ij} is the weight of the edge between i and j in the projected one mode network. This enables us to generate a weighted projected one mode network from the bipartite network in an efficient way. Further, we avoid the loss of information for a node of degree one in the secondary set S . An illustration of the weighted projection method we use appears in Fig. 2.4.

We have programmed our projection algorithm in C^{++} for compatibility with the implementations we have of the Infomap and Louvain algorithms. We start by reading the bipartite network edges as a pair of nodes, the first from P and the second from S . The labels on the nodes in this dataset do not have to be numbers, they can be post codes, book serials, bank card numbers, names of social networks or even names of people. Then, we use special techniques in C^{++} that affect the efficiency of the projection method [42]. Using a C^{++} container called Mapvector which requests a key and a value, we choose each key as an element of S and its value to be a vector of nodes in P to which it is adjacent. Then, we create pairs in a one mode network and store the result in container called “Multiset”.

To solve the labeling issue we use a mapping between strings and integers and generate new numbers that represent list of pairs with the links between nodes. After the projection we will have only (number, number) pairs which is exactly what Infomap requires. However, another issue arises here, that of losing the initial strings/labels of nodes. When we generate the final network picture we will see only links between these numbers but without any label on them. Therefore, we use the Pajek format because when declaring the nodes, we can also give a label for each node. We then input the projected network data into the Infomap² (and Louvain³) algorithms.

The pseudocode is given in Algorithm 1. We illustrate the intermediate steps of Algorithm 1 as it applies to the toy network in Fig. 2.4.

²Infomap available for download on the link: www.mapequation.org/.

³Louvain available for download on the link: <https://sites.google.com/site/findcommunities/>.

Table 2.1 Time complexity for algorithms on bipartite networks with n nodes and m edges

Algorithm	Order
LPAb	$O(m)$
LPAb+	$O(m \log^2 n)$
Algorithm 1	$O(m^2) + O(m)$

Example 2.1 In Fig. 2.4 we have $P = \{1, 2, 3, 4, 5, 6, 7\}$, $S = \{A, B, C, D, E\}$. The operation of Algorithm 1 on this network at the following lines will be:

```

4  Mapvector(string) = [(A, 1)(A, 2)(A, 4)(A, 5)(B, 1)(B, 2)(B, 4)(B, 6)
   (C, 3)(C, 4)(C, 6)(D, 2)(D, 5)(D, 7)(E, 4)]
13 Mapvector[i, j] = [(A, 1 2 4 5)(B, 1 2 4 6) (C, 3 4 6) (D, 2 4 7) (E, 4)]
25 Multiset[i, j] = [(1, 2)(1, 4)(1, 5)(2, 4)(2, 5)(4, 5)
   (1, 2)(1, 4)(1, 6)(2, 4)(2, 6)(4, 6)
   (3, 4)(3, 6)(4, 6)
   (2, 5)(2, 7)(5, 7)
   (4, 4)]

```

We can compute the time complexity for the whole operation starting from converting bipartite networks to weighted unipartite networks followed by clustering them using either algorithm. The reason the projection method is also applied to the Louvain algorithm is to be able to compare the performance of Infomap with that of Louvain. The complexity for both Infomap and Louvain is $O(m)$ where m is the number of the edges in G . Our projection takes $O(m^2)$. Table 2.1 summarizes the complexity of the integrated algorithm. Although the efficiency of our algorithm is comparable with those applying bimodularity, it is not as good as those employing label propagation, as Table 2.1 shows. The running time needs to be improved.

To evaluate the quality of community detection in a bipartite network using Algorithm 1, we look to examples where it is possible to extract some ground truth. There is no suite of existing benchmark bipartite networks for testing purposes comparable to the LFR benchmarks [27] for one mode networks. The most-studied bipartite network is the very small “Southern women” network and it has been used as a de-facto benchmark for testing community detection algorithms, both for bipartite graphs (obviously not Infomap, though) and for the projection onto P .

2.4.2 Benchmark “Southern Women” Dataset

The “Southern women” network collected by Divas et al. [12] has become a benchmark for testing community detection algorithms on bipartite networks. This network has 18 women (who form the primary set P) who attended 14 different events (the secondary set S). An edge exists between two women for each event they attend

Algorithm 1: Weighted projection method for bipartite network integrated with Infomap or Louvain algorithm.

Require: A bipartite network.

```

1: initialization
2: while end of dataset not reached do
3:   read each pair from dataset
4:   store pairs in Mapvector[string]
5: end while
6: Find common neighbors:
7: for all  $i = 1$  end of mapvector'keys' do
8:   print(i)
9:   for all  $j = 1$  end of mapvector'value' do
10:    print(j)
11:   end for
12: end for
13: return :Mapvector[i,j]
14: Create pairs for one mode network:
15: for all  $i = 1 \rightarrow$  end of Mapvector[ $i, j$ ] do
16:   if size of commonneighbor = 1 "self loop" then
17:    insert the duplicate [ $i, i$ ] into multiset
18:   else
19:    for  $i = 1 \rightarrow$  end of Mapvector - 1 do
20:      for  $j = i + 1 \rightarrow$  end of commonneighbors do
21:        insert the pair [ $i, j$ ] into multiset
22:      end for
23:    end for
24:   end if
25: end for
26: return : Multiset[i, j]
27: Create the associated pairs of vertices and store them in Pajek format from this Multiset:
28: for  $i = 1 \rightarrow$  end of Multiset[ $i, j$ ] do
29:   store vertices in string variable  $\leftarrow$  List of vertices with its Labels
30: end for
31: while the end of Multiset not reached do
32:   currentpair = *begin of Multiset
33:   if the both pair numbers are the same then
34:    print edges[ $i, i$ ]
35:    count (duplicate pairs) /* to avoid the redundant pairs */
36:   else
37:    save current pair
38:    count to list of edges string
39:    erease current pair from Multiset /* to enhance the computational time */
40:   end if
41: end while
42: store edges in string variable  $\leftarrow$  list of the edges with weights
43: Reading input network from string variable rather than from screen
44: while string variable not empty do
45:   read the input from weighted projection approach as Pajek format
46: end while
47: process the normal Infomap or Louvain algorithm

```

Table 2.2 The Southern women network: Number of communities of women detected by different algorithms

Algorithm	Quality function	Network applied to	Modules in P
Alzahrani et al. [5]	Modularity	Weighted projection	2
Guimera et al. [22]	Modularity	Weighted projection	2
Crampes and Plantie [10]	Bimodularity	Bipartite	3
Barber [6]	Bimodularity	Bipartite	4
Liu and Murata [31]	Bimodularity	Bipartite	4
Alzahrani et al. [5]	Map equation	Weighted projection	4

together. Most studies conducted before 2003 identify two (sometimes overlapping) communities of women while one identifies three communities [17]. In many studies, members within each community are further partitioned into core or peripheral members. More recent studies using bimodularity find more communities (3 and 4). Consequently, at least two communities are expected. In Table 2.2, we list the community numbers found in the Southern women dataset by the more recent bipartite network algorithms described in Sect. 2.3 and by our implementation of projection in Infomap and Louvain.

We compare our results for the Southern women network with results in the literature, in more detail. Using Infomap, we have community A consisting of Evelyn and Theresa (women 1 and 3, respectively), community B consisting of Katherine and Nora (women 12 and 14, respectively), and two others $C = \{8, 9, 16, 17, 18\}$ and $D = \{2, 4, 5, 6, 7, 10, 11, 13, 15\}$, as shown in Fig. 2.5. Our groups A and B consist of women frequently identified as core members of each of the two communities found in earlier studies. By contrast, Barber’s two smaller communities consist of women who tended to be identified as peripheral members of each of the two communities found in earlier studies [17]. Barber also tested the success of his partition into four communities, found using the maximum bipartite modularity (as described in Sect. 2.3), as a partition in the corresponding *unweighted* projection network, and found it to have negative modularity [6]. As this is worse than considering the women as a single community, it further supports our use of the *weighted* projection network. Guimera et al. [22] found only two communities of women (red and blue) whether modularity on the unweighted projection, the weighted projection or bipartite modularity was used. They found the communities were inaccurate with unweighted projection, but identical and in agreement with supervised results in [17] for the other two methods. The total number of edges in the Southern women network after weighted projection is 139 edges. Our community A (Evelyn and Theresa) has internal edge weight 7 and lies inside the red group, while our community B (Katherine and Nora) has internal edge weight 5 and lies inside the blue group. These two “core” strong small communities are not detected by the modularity based algorithm, probably because their edge numbers fall below the resolution limit of modularity, which in this case is 12 (since $11 < \sqrt{139} < 12$). By comparison the

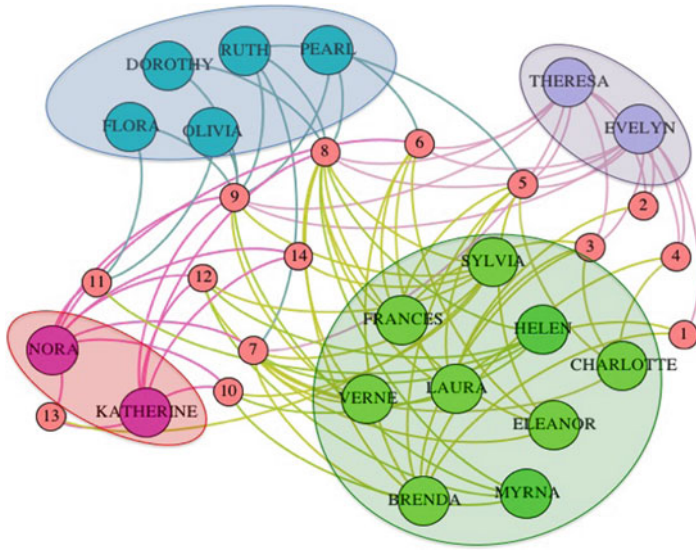


Fig. 2.5 The four communities of women found in the Southern women network. *Red nodes* represent S , the events the women attended, and the four other colors represent four communities within P , with nodes labelled by first name [5]

2 communities found by our projection input into Louvain have 45 and 33 internal edges. This demonstrates that, in this benchmark bipartite case, Louvain is subject to the resolution limit for modularity but Infomap is not.

2.5 Case Studies

Application of Infomap to the small and much analysed “Southern women” bipartite network shows that the communities detected represent meaningful associations between the women grouped together. In this section we continue to apply Infomap and Louvain to weighted projections of three larger bipartite networks as case studies. We demonstrate that Infomap produces meaningful communities representing some sort of ground truth, and does so better than Louvain. The case studies are presented in order of increasing size, as each highlights a different feature of Infomap community detection.

2.5.1 Noordin Top Terrorist Network

The Noordin Top terrorist group data linking individuals with relationships or affiliations first appeared in [25]. The ties or links between actors represent one or more common affiliations or relationships. Common attendance of actors at events was

Table 2.3 Communities in the Noordin Top terrorist network [4]

Algorithm	Communities	Sizes
Louvain	4	29, 16, 15, 14
Infomap	5	25, 30, 12, 4, 3

inferred from their mention together in public reports in newspapers and elsewhere. The data were coded as network data by Naval postgraduate students and the information was published in 2012 in [14]. We work with the cleaned affiliation subnetwork and thank Assoc. Prof. Murray Aitkin for providing it. It forms a bipartite network with 79 actors and 45 events (affiliations), classified into six categories (Operations, Logistics, Organizations, Training, Finance, Meeting). We excluded the actors who did not present at any of the 45 events.

In [2] the Bayesian latent class model of [3] (see Sect. 2.3.3) is applied directly to this terrorist network for $K = 1, \dots, 4$. The researchers find the $K = 3$ model fits best and use an actor's degree to assign them to a community. Their first group consists of two important leaders and planners (Noordin Top and Azahari Husin), and they conclude that the other two groups are: the "footsoldiers"; and the intermediaries who meet the planners and train the footsoldiers.

Weighted projection of the Noordin Top bipartite network onto the actor set P determines a network with $|E_P| = 759$ edges in total weight. Using the Infomap algorithm we found 5 communities and using Louvain we found 4 communities, see Table 2.3. The modularity resolution limit for this network is $\lfloor \sqrt{759} \rfloor = 27$. Therefore, a community with strong ties and $\ll 27$ edges may not be detected by modularity based methods. The 5 communities found by the Infomap algorithm are displayed in detail in Fig. 2.6.

The smallest Louvain community (14 actors) wholly contains the third Infomap community (12 actors), and we regard them as essentially equivalent. The largest Louvain community (of 29 actors) contains 23 of the 25 actors belonging to the largest Infomap community. It also contains the smallest Infomap community (a clique of 3 actors with weighted edge sum 6). The second small Infomap community (a clique of 4 actors with weighted edge sum 6) has three actors in the largest Louvain community and one in the second largest Louvain community. *Essentially, Infomap detects three communities inside the largest Louvain community.* The two small clique communities are half an order of magnitude smaller than the modularity resolution limit. This is a real-world illustration of the phenomenon illustrated theoretically in Fig. 2.2.

Consequently, to test the communities found for meaningfulness, we concentrate on the structure found by the Infomap algorithm.

Community 4 contains actors Abdul Rauf, Imam Samudra, Apuy and Baharudin Soleh. Community 5 contains actors Enceng Kurnia, Anif Solchanudin and Salik Fridaus. These two small cliques have no recorded direct links between them, nor does Community 5 have any recorded direct links with Community 3. Identifying these small clique communities in the original bipartite network described in [46] recovers very meaningful link information. For instance, Anif Solchanudin and Salik Fridaus

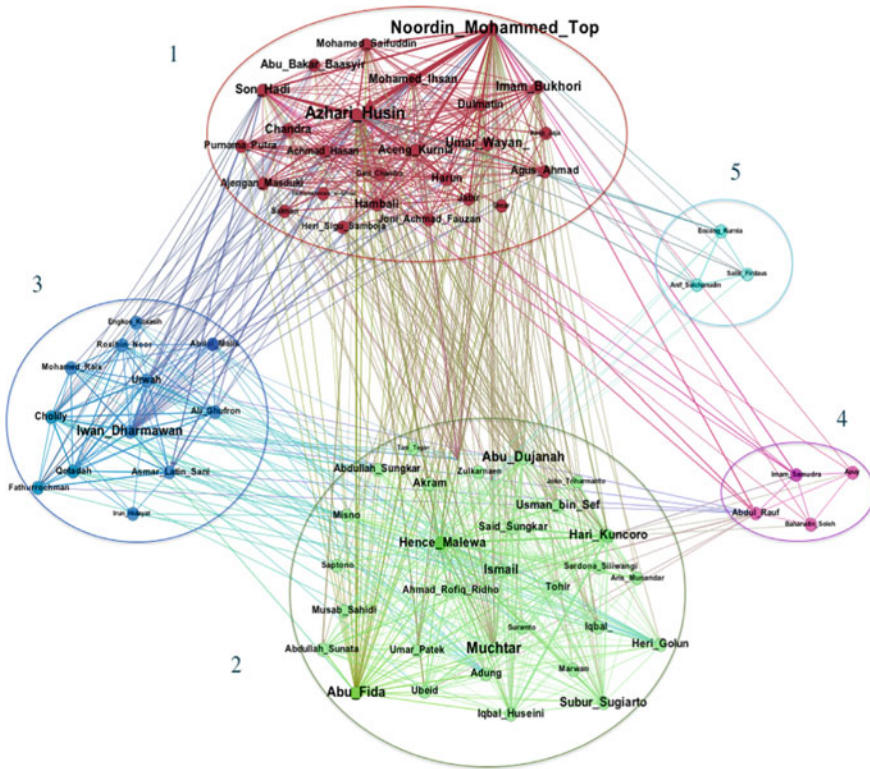


Fig. 2.6 Noordin Top terrorist network actor communities found using Infomap: Community 1 (red, top, 25 actors), Community 2 (green, bottom, 30 actors), Community 3 (purple, left, 12 actors), Community 4 (pink, below right, 4 actors), Community 5 (blue, above right, 3 actors)

were trained together to be suicide bombers for Bali Bomb II in 2005. Community 4 also reflects useful information. Abdul Rauf, Imam Samudra and Apuy came from the same organization, Ring Baten, while Apuy and Baharudin Soleh were involved directly in the Australian Embassy bombing in 2004. These two smallest communities are new structure, not found by the Louvain algorithm or in [2], and are significant from a defence analysis perspective.

In [4] we related Fig. 2.6 back to the 6 categories of events, as was done in [2] for only 3 communities, and listed the “Top Ten” actors by four different centrality measures. Community 1 contains the two principal leaders and planners (Noordin Top and Azhari Husin). In total, 8 of the 14 actors appearing in the Top Ten lists are in Community 1. The most significant common property of this group is that 17 out of 25 of its actors were affiliated to the same Organization (Jemaah Islamiyah, a transnational Southeast Asian militant Islamist terrorist organisation linked to Al-Qaeda), and we can conclude it is the most significant community.

2.5.2 NSW Crime

This publically available Australian crime data from the state of New South Wales (NSW) was published in 2012 [37]. It was collected by the NSW Bureau of Crime Statistics and Research from January 1995 to 2009, and it provides rich information about every crime that occurred in each month, categorised by offence type. There are 21 offence categories; some of these categories have subcategories that are related to the main category of the offences. For instance, “Homicide”, as a category of offence, has four subcategories (Murder, Attempted Murder, Accessory to Murder and Manslaughter) that all relate one way or another to the main category. The underlying social network of offenders is reflected in the reported crimes.

The data reports the crime according to the local government area (LGA) it was committed in. There are 155 LGAs in NSW. The bipartite network we extract has as node sets the offence categories and the LGAs that they were committed in, and has $m = 8761$. We are interested in identifying where similar patterns of crime have occurred, and which are the more dangerous areas, so P is the LGAs and S is the categories of offence. Weighted projection onto P results in an extremely dense network with $|E_P| = 3,478,084$ edges in total weight.

We applied both the Infomap and Louvain algorithms to the weighted projection on P . The Louvain algorithm did not determine any community structure at all. Consequently it is of no use for analytic purposes. However the Infomap algorithm found 2 communities of LGAs, one containing 82 LGAs and the other containing 73 LGAs. We expect there is more frequent connection between some subset of crimes for Community 1 of LGAs versus the more frequent connection between some other subset of crimes for Community 2. The modularity of this structure is higher than that for a single community, see Table 2.4, indicating it is a better structure, so the modularity-maximising Louvain algorithm should have found more than one community.

In fact it is somewhat surprising that so few communities were found. The number of internal edges in the larger community found by Infomap is 112,374, almost two orders of magnitude greater than the modularity resolution limit of $\lfloor \sqrt{3,478,084} \rfloor = 1,864$. A possible explanation is that *the communities are very weak, having a high average mixing parameter*, and so are difficult for any algorithm to detect.

However, when the LGAs in NSW are mapped and coloured according to community, a very strong geographical divide is visible. It provides a dramatic explanation of the community partition found by Infomap. Generally speaking, Community 1 includes the more populated LGAs and Community 2 includes the majority of rural and “Outback” LGAs. The 38 LGAs in the main metropolitan area, Sydney, are all in Community 1.

Table 2.4 Comparison of algorithm performance on NSW crime network [4]

Algorithm	Communities	Sizes	Modularity
Louvain	1	155	0
Infomap	2	82, 73	0.026

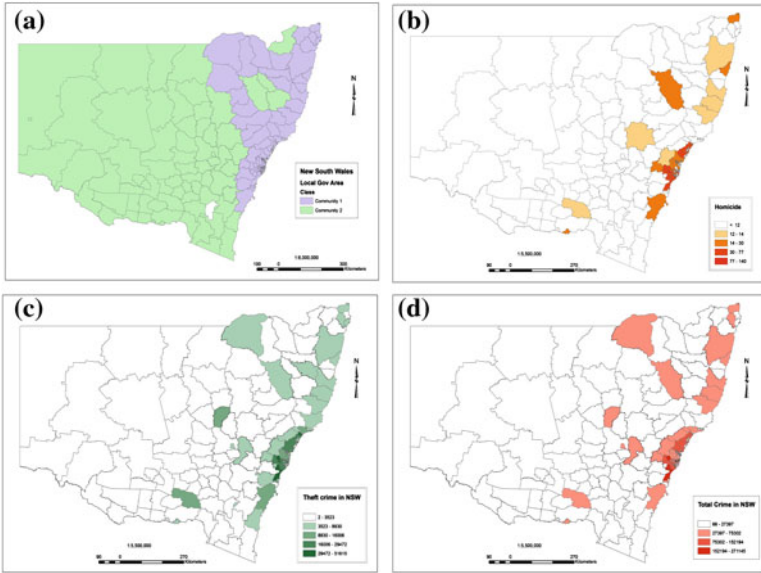


Fig. 2.7 Map of NSW local government areas and related crimes. **a** The 2 communities of LGAs found by Infomap: Community 1 contains 82 LGAs and Community 2 contains 73 LGAs. The unclassified area is the Australian Capital Territory, which is not part of NSW. Underlying crime statistics are also mapped by LGA: **b** Homicide rate; **c** Theft rate and **d** Total crime rate [4]

Analysis of the underlying crime statistics by LGA shows that for homicide (Fig. 2.7b), 90% of the shaded LGAs occur in Community 1; for theft (Fig. 2.7c) 85% of the shaded LGAs occur in Community 1 and for total crime rate (Fig. 2.7d), 86% of the shaded LGAs occur in Community 1. The correlation coefficient of the crime rates between the two communities is 0.992. Deeper analysis of this network will be undertaken elsewhere.

2.5.3 Internet Movie Database Network

The Internet Movie Database (IMDB) is downloadable from [26]. We thank Dr Tiago Peixoto for the cleaned dataset from [40] that we use here. The dataset includes details about internet media and actors in them from different perspectives such as country and year of production, genre, language and rating. The “Internet Movie” term covers a range of film types, including movies, video shows, TV shows and video games; the actors are the cast members.

We are interested in the bipartite network formed from this database, where films form the primary set P and actors who have acted in a film listed in P form the secondary set S . Initially we have 275,805 actors who participated in 96,982 films. The number of edges is $\approx 1,812,697$, each edge represents an actor appearing in a film. The actors and films with degree $k \leq 1$ have been removed since they provide no

significant information on the overall network structure, giving $|P| = 96,881$. The corresponding weighted edge number is $|E_P| = 18,772,909$ and $\lfloor \sqrt{18,772,909} \rfloor = 4,332$.

The MDL stochastic block model (see Sect. 2.3.3) was applied to the whole network directly in [40] and $K = 332$ communities found, which, remarkably, perfectly reflected the underlying bipartiteness, with 165 communities entirely in P and 167 entirely in S . Note that $n = |P| + |S| = 372,787$ so $\lceil \sqrt{n} \rceil = 611$ and the maximum number of communities this algorithm can detect in the whole network is of this order.

Clustering the weighted projected network using Infomap results in 682 clusters of films in P . When we apply Louvain, only 64 clusters in P result. However, checking the four levels of the Louvain algorithm shows decreasing cluster numbers (level 0: 96,881 nodes; level 1: 528 nodes; level 2: 80 nodes; level 3: 64 nodes). In accordance with the Erratum [29], to avoid the resolution limit for modularity we take 528 as the number of film communities found by Louvain.

Thus, it seems likely that the 165 film clusters in P found by [40] is an underestimate, and the MDL stochastic block model suffers from its resolution limit in this case.

In Fig. 2.8 we plot the log degree distribution of P and the distribution of community sizes found by Infomap in P . Both demonstrate a clear heavy tail. The *community*

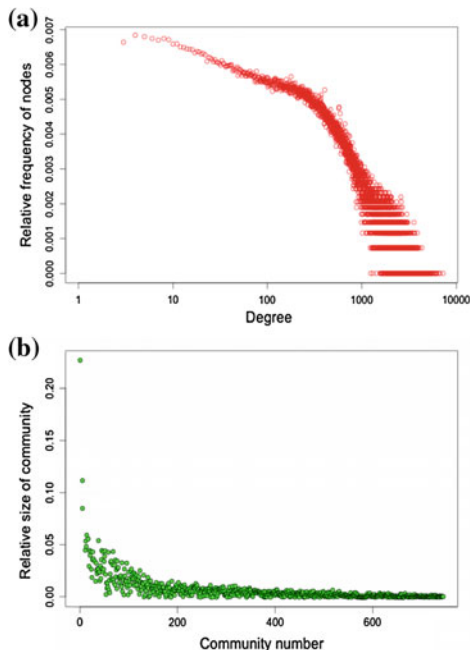


Fig. 2.8 IMDB projected film network: **a** Node degree distribution (log scale) and **b** Infomap community sizes, relative to network size

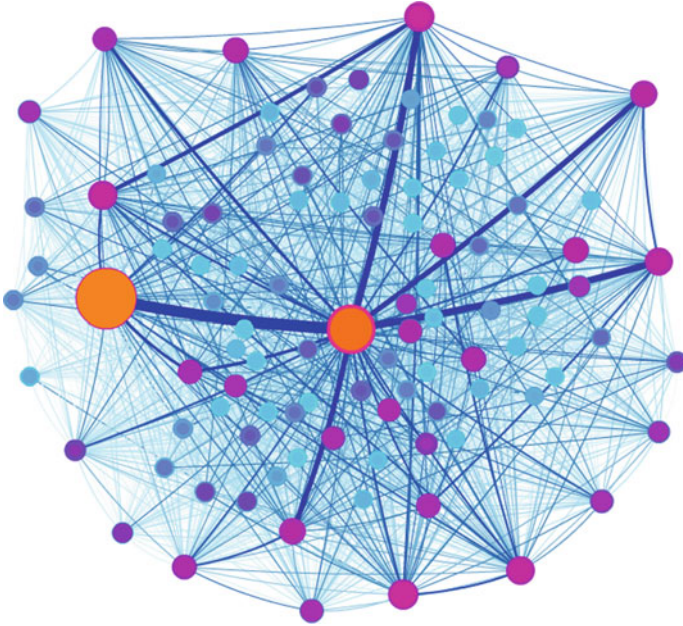


Fig. 2.9 IMDB projected film network: The largest 100 out of 682 clusters for the weighted projected network found using Infomap. It shows two giant clusters, the more central one with 10,240 nodes and the other with 22,727 nodes

size distribution is immune to any resolution limit: the smallest communities in P have 2 nodes. We conclude that in this projected network the hierarchical structure is well-defined and the communities are well separated.

For clarification, in Fig. 2.9 we show the largest 100 clusters as supernodes that clarify the structure of the projected film network. Two of the Infomap clusters are giant components; the first one has 22,727 nodes, all of which are the same film type (movie), and includes almost a quarter of P , and the second has 10,240 nodes all of which are movies as well. We checked the IMDB data briefly to see if these two clusters make sense, and they do indeed represent some ground truth. For example in the second giant cluster, almost all movies all have the same country of production (USA) and genre classification (Drama). The top 5 *hubs* (nodes with highest degree) [33] belong to these two giant components.

The second giant cluster is more central, even though it has fewer nodes than the other, for intrinsic reasons. The betweenness centrality for nodes in the second giant cluster (the number of shortest paths between node pairs in the network that pass through that node), is higher than for nodes belonging to the first giant cluster. The top three nodes for betweenness (the highest is for the 2009 movie “Never” (Part 1) and the second highest is for the 2008 movie “Around June”) and the largest hub (“Around June” with $k = 7251$) are in the second largest component.

2.6 Conclusions and Further Work

In this chapter we have reviewed a collection of community detection algorithms, including variants specifically designed for bipartite networks, that have previously been used to cluster bipartite networks. Modularity based algorithms suffer from a well-known resolution limit but the best-performing algorithm for large networks, the random-walks based Infomap, cannot be applied to a bipartite network directly.

For four bipartite networks of increasing size, we have applied Infomap to the weighted network projected onto the primary node set and compared its performance with the most popular modularity based algorithm, Louvain, and with other algorithms reported in the literature. Evaluation of detected clusters has shown that the clusters found using Infomap do embody meaningful information about the ground truth of hierarchical structure within network. Infomap can detect meaningful small communities such as cliques with sizes below the resolution limit of modularity based algorithms (the Southern women and Noordin Top terrorist networks). Infomap can detect weak large clusters better than Louvain at the upper limit of mixing coefficient (NSW crimes network). Infomap can detect a full hierarchy of clusters, that is, with no resolution limit, when they are well-defined (IMDB network).

There are number of reasons that a random walks based algorithm should be considered for community detection in bipartite networks. First, as has been our focus in this paper, it is frequently the case that the principal interest in the network is in the clustering within only one of the node sets. In this case, we believe we have shown a clear advantage in applying Infomap to detect meaningful communities in the primary projected network.

More generally, Infomap has the best performance against the LFR benchmark, so it is worthwhile to try to adapt it to bipartite networks. Moreover, the lack of existence of a benchmark for clustering algorithms on bipartite networks underlines the flexibility for researchers to employ new approaches that might suit the bipartite framework. A further reason that Infomap should be considered for bipartite networks is that it provides the sense of ground truth behind the cluster formation.

We intend to project the two sets P and S of the bipartite network in parallel, cluster them separately using the random walks based algorithm and merge their results within the bipartite network. Finally we plan to compare these bipartite communities with those clusters found by modularity-based bipartite clustering and those using multi assignment clustering.

One important observation made during the detailed study in this chapter is that nodes of the primary set might in fact belong to more than one community when the information from the secondary set is taken into account. Investigation of overlapping communities is possible future work.

Acknowledgments We are very grateful to Assoc. Prof. Murray Aitken for supplying us with the cleaned affiliation network data for the Noordin Top terrorist network; to Dr Tiago Peixoto for supplying us with the cleaned IMDB database; and to Assoc. Prof. Chris Bellman and Ms Sarah Taylor for assistance in using the ArcGIS mapping software on our clustered NSW crime network. The first author would like to thank the Ministry of Finance of Saudi Arabia for supporting his research. The work of the second author was partly supported by Department of Defence of Australia Agreement 4500743680. This work forms part of the PhD thesis of the first author, taken under the supervision of the second author.

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* **466**(7307), 761–764 (2010)
2. Aitkin, M., Vu, D., Francis, B.: Statistical modelling of a terrorist network (2013)
3. Aitkin, M., Vu, D., Francis, B.: Statistical modelling of the group structure of social networks. *Soc. Netw.* **38**, 74–87 (2014)
4. Alzahrani, T., Horadam, K.J.: Analysis of two crime-related networks derived from bipartite social networks. In: Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pp. 890–897. IEEE (2014)
5. Alzahrani, T., Horadam, K.J., Boztas, S.: Community detection in bipartite networks using random walks. Proceedings of CompleNet 2014. Springer Studies in Computational Intelligence, vol. 549, pp. 157–165 (2014)
6. Barber, M.J.: Modularity and community detection in bipartite networks. *Phys. Rev. E* **76**(6), 066102 (2007)
7. Barber, M.J., Clark, J.W.: Detecting network communities by propagating labels under constraints. *Phys. Rev. E* **80**(2), 026129 (2009)
8. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.* **7**, 6–17 (2002)
9. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**, P10008 (2008)
10. Crampes, M., Plantie, M.: A unified community detection, visualization and analysis method (2013). arXiv preprint [arXiv:1301.7006](https://arxiv.org/abs/1301.7006)
11. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *J. Stat. Mech.: Theory Exp.* **2005**, P09008 (2005)
12. Davis, A., Gardner, B.B., Gardner, M.R.: Deep South: A Social Anthropological Study of Caste and Class. University of Chicago Press, Chicago (1941)
13. Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**(1), 016105 (2009)
14. Everton, S.F.: Disrupting Dark Networks. Cambridge University Press, Cambridge (2012)
15. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010)
16. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **104**(1), 36–41 (2007)
17. Freeman, L.C.: Finding social groups: a meta-analysis of the southern women data. In: Breiger, R., Carley, K.M., Pattison, P. (eds.) *Dynamic Social Network Modeling and Analysis*, pp. 39–97. National Academies Press, Washington (2003)
18. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99**(12), 7821–7826 (2002)
19. Grunwald, P.D., Myung, I.J., Pitt, M.A.: *Advances in Minimum Description Length: Theory and Applications*. MIT Press, Cambridge (2005)
20. Guillaume, J.L., Latapy, M.: Bipartite structure of all complex networks. *Inf. Process. Lett.* **90**(5), 215–221 (2004)

21. Guimera, R., Sales-Pardo, M., Amaral, L.S.A.N.: Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* **70**(2), 025101 (2004)
22. Guimera, R., Sales-Pardo, M., Amaral, L.S.A.N.: Module identification in bipartite and directed networks. *Phys. Rev. E* **76**(3), 036102 (2007)
23. Huffman, D.A.: A method for the construction of minimum-redundancy codes. *Proc. IRE* **40**(9), 1098–1101 (1952)
24. Hu, Y., Chen, H., Zhang, P., Li, M., Di, Z., Fan, Y.: Comparative definition of community and corresponding identifying algorithm. *Phys. Rev. E* **78**(2), 026121 (2008)
25. International Crisis Group: Terrorism in Indonesia: Noordin's Networks. Asia Report no. 114, Brussels, Belgium (2006)
26. Internet Movie Database original database [Online]
27. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
28. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**(5), 056117 (2009)
29. Lancichinetti, A., Fortunato, S.: Erratum: Community detection algorithms: A comparative analysis. *Phys. Rev. E* **80**, 056117 (2009) (*Phys. Rev. E*, **89**(4), 049902 (2014))
30. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society, Providence (2009)
31. Liu, X., Murata, T.: An efficient algorithm for optimizing bipartite modularity in bipartite networks. *JACIII* **14**, 408–415 (2010)
32. Mukherjee, A., Choudhury, M., Ganguly, N.: Understanding how both the partition of a bipartite network affect its one-mode projection. *Phys. A* **390**(20), 3602–3607 (2011)
33. Newman, M.E.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
34. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**(3), 036104 (2006)
35. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004)
36. Nishikawa, T., Motter, A.E., Lai, Y.C., Hoppensteadt, F.C.: Heterogeneity in oscillator networks: are smaller worlds easier to synchronize? *Phys. Rev. Lett.* **91**(1), 014101 (2003)
37. NSW Bureau of Crime Statistics and Research. Dataset [Online]. NSW Crime data (2008)
38. Orman, G.K., Labatut, V., Cherifi, H.: On accuracy of community structure discovery algorithms (2011). arXiv preprint [arXiv:1112.4134](https://arxiv.org/abs/1112.4134)
39. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
40. Peixoto, T.P.: Parsimonious module inference in large networks. *Phys. Rev. Lett.* **110**, 148701 (2013). (Erratum. *Phys. Rev. Lett.* **110**(16), 169905 (2013))
41. Pons, P., Latapy, M.: Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10**(2), 191–218 (2006)
42. Preiss, B.R.: Data Structures and Algorithms with Object-Oriented Design Patterns in C++. Wiley Press, New York (1997)
43. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA* **101**(9), 2658–2663 (2004)
44. Raghavan, U.N., Albert, R.K., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
45. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proc. Natl. Acad. Sci.* **101**(9), 2658–2663 (2004)
46. Roberts, N., Everton, S.F.: Strategies for combating dark networks. *J. Soc. Struct.* **12**, 1–32 (2011)
47. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**(4), 1118–1123 (2008)
48. Zhou, T., Lu, L., Zhang, Y.C.: Predicting missing links via local information. *Eur. Phys. J. B* **71**(4), 623–630 (2009)

Complex Systems and Networks

Dynamics, Controls and Applications

Lu, J.; Yu, X.; Chen, G.; Yu, W. (Eds.)

2016, VIII, 482 p. 196 illus., 158 illus. in color.,

Hardcover

ISBN: 978-3-662-47823-3