

# Group detection in complex networks: An algorithm and comparison of the state-of-the-art

Lovro Šubelj\*, Marko Bajec

*University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, SI-1000 Ljubljana, Slovenia*

---

## Abstract

Complex real-world networks commonly reveal characteristic groups of nodes like communities and modules. These are of value in various applications, especially in the case of large social and information networks. However, while numerous community detection techniques have been presented in the literature, approaches for other groups of nodes are relatively rare and often limited in some way. We present a simple propagation-based algorithm for general group detection that requires no apriori knowledge and has near ideal complexity. The main novelty here is that different types of groups are revealed through an adequate hierarchical group refinement procedure. The proposed algorithm is validated on various synthetic and real-world networks, and rigorously compared against twelve other state-of-the-art approaches on group detection, hierarchy discovery and link prediction tasks. The algorithm is comparable to the state-of-the-art in community detection, while superior in general group detection and link prediction. Based on the comparison, we also discuss some prominent directions for future work on group detection in complex networks.

*Keywords:* complex networks, group detection, hierarchy discovery, label propagation, clustering.

*PACS:* 89.75.Hc, 89.75.Fb, 87.23.Ge, 89.20.Ff

---

## 1. Introduction

Complex networks of real-world systems commonly reveal groups of nodes with characteristic connection pattern [1] (e.g., densely connected groups known as communities [2]). These correspond to people with common interests in social networks [3] or classes with the same information signature in software networks [4]. Characteristic groups of nodes provide an important insight into the structure and function of real-world networks [5], while group detection also has numerous practical applications, including epidemic outbreak prevention [6], viral marketing [7], software package prediction [8] and compression [9].

Despite an outburst of community detection algorithms in the last decade [10, 1], approaches for other groups of nodes are relatively rare and often limited (e.g., demand some apriori knowledge about the network structure). Thus, we here propose a general group detection algorithm based on the label propagation framework in [4, 11] that requires no apriori knowledge. Analysis in the paper confirms that the proposed algorithm is at least comparable to the current state-of-the-art, while its complexity is near ideal. The main novelty of the paper is else a simple hierarchical refinement procedure that enables straightforward discovery of different types of groups. The paper also includes a detailed empirical comparison of a larger number of state-of-the-art approaches for group detection that may be of an independent interest.

The rest of the paper is structured as follows. First, Section 2 gives preliminary discussion on groups of nodes in real-world networks. Next, we introduce the group detection algorithm proposed in the paper in Section 3. Rigorous analysis on synthetic and real-world network appears in Section 4, whereas detailed comparison with the state-of-the-art on group detection, hierarchy discovery and link prediction tasks is presented in Section 5. Section 6 concludes the paper and gives some prominent directions for future work.

---

\*Corresponding author. Tel.: +386 1 476 81 86.

*Email addresses:* [lovro.subelj@fri.uni-lj.si](mailto:lovro.subelj@fri.uni-lj.si) (Lovro Šubelj), [marko.bajec@fri.uni-lj.si](mailto:marko.bajec@fri.uni-lj.si) (Marko Bajec)

## 2. Background

Let the network be represented by a simple undirected graph  $G(V, L)$ , where  $V$  is a set of nodes in the network,  $|V| = n$ , and  $L$  is the set of links,  $|L| = m$ . Next, let  $\Gamma_i$  be the set of neighbors of node  $v_i$ ,  $v_i \in V$ , and  $\Delta_i$  the number of links between the nodes in  $\Gamma_i$ . Last, let  $k_i$  be the degree of  $v_i$ ,  $|\Gamma_i| = k_i$ , and let  $\langle k \rangle$  be the average degree in the network.

### 2.1. Groups in real-world networks

The present paper is concerned with groups of nodes with characteristic connection pattern that appear in complex real-world networks [12]. While these could be defined in various ways, we adopt two types of groups that have been most popular in the recent literature [10, 1].

First, we consider communities [2] (also link-density community [13]) that, e.g., represent groups of people with common interests in social networks [3]. Community is defined as a (connected) group of nodes with more links towards the nodes in the group than to the rest of the network [14].

Second, we consider sparse modules [15] (also link-pattern community [13] and other [16]) that in, e.g., software networks correspond to classes with the same function [4]. Module is defined as a (possibly) disconnected group of nodes with more links towards common neighbors than to the rest of the network [4].

Definition of a module is rather similar to the concept of structural or regular equivalence [17, 18] (i.e., blockmodels), although not equivalent. Note that communities can in fact be considered under the definition of modules (most authors have indeed adopted this stance [12, 16]), however, there exist important differences between the two [4] (e.g., connectedness).

### 2.2. Node and network clustering

Based on the above, a group detection approach could gain from differentiating between communities and modules. Note that the definition of a community implies locally dense structure, while no such tendency appears in the definition of a module. Density of node's neighborhood is usually measured by the node clustering coefficient  $c$  [19] defined as

$$c_i = \frac{\Delta_i}{\binom{k_i}{2}}, \quad (1)$$

while  $C = \frac{1}{n} \sum_i c_i$  is the network clustering coefficient [19, 20].

However, denominator in Eq. (1) introduces biases into the definition of clustering, since  $\binom{k_i}{2}$  often cannot be reached due to fixed degree distribution [21]. The latter is particularly apparent in degree disassortative networks [22], which most real-world networks in fact are. Thus, an alternative definition denoted degree-corrected clustering coefficient  $d$  [21] has been proposed as

$$d_i = \frac{\Delta_i}{\omega_i}, \quad (2)$$

where  $\omega_i$  is the maximum possible number of links among  $\Gamma_i$  and  $D = \frac{1}{n} \sum_i d_i$ .

Thus, in the presence of communities, one can expect dense network structure with  $D \gg 0$ . On the other hand, different configurations of modules (e.g., bipartite, multi-partite or star-like structures) imply sparser networks with  $D \approx 0$ . For example, most prominent modules are found in two-mode networks, where  $D = 0$ . It should, however, be stressed that a module structure does not necessarily mean  $D = 0$ , since a tripartite configuration of modules obviously has  $D > 0$ .

## 3. The algorithm

The proposed algorithm for group detection is based on the label propagation framework [4, 11] that we introduce next. Due to simplicity, the framework is presented for the case of simple undirected graphs, although a generalization to weighted multigraphs is straightforward.

### 3.1. Group detection by propagation

Let  $g_i$  be an unknown group label of node  $v_i$  and let  $\mathcal{G}_i$  a group of nodes with label  $g_i$ . Propagating labels between the nodes was first proposed for community detection [23]. Initially, each node is labeled with a unique label as  $g_i = i$  (i.e., put in a separate group  $\mathcal{G}_i$ ). Then, at each iteration of the algorithm, each node adopts the label shared by most of its neighbors (ties are broken uniformly at random). Hence,

$$g_i = \operatorname{argmax}_g \sum_{v_j \in \Gamma_i} \delta(g_j, g), \quad (3)$$

where  $\Gamma_i$  is the set of neighbors of node  $v_i$  and the  $\delta$  is a Kronecker delta. Due to the presence of numerous links within the communities, relative to the number of links towards the rest of the network, nodes in communities form a consensus on some label  $g$  after only a few iterations. Thus, when the process converges, disconnected groups  $\{\mathcal{G}\}$  are classified as communities. Due to very fast structural inference of label propagation, the algorithm exhibits near linear complexity, while the expected number of iterations on a network with a billion links is only 113 [24].

The approach can be significantly improved by adopting also node preferences [25]. Preferences adjust the propagation strength of the nodes and force the label propagation process towards a desirable group partition  $\{\mathcal{G}\}$ . Let  $f_i$  and  $b_i$  be the preferences of node  $v_i$  [4] (see below). Then, Eq. (3) is rewritten to

$$g_i = \operatorname{argmax}_g \sum_{v_j \in \Gamma_i} f_j b_j \cdot \delta(g_j, g). \quad (4)$$

Preferences  $f$  correspond to defensive label propagation [24] that increases the propagation strength from the core of each group or, equivalently, decreases the strength of its border. This forces the algorithm to more gradually reveal the underlying structure and improves group detection in real-world networks [24]. The core and the border of each group are modeled by employing random walks (see Alg. 1).

Updates of nodes' labels in Eq. (3) and Eq. (4) occur sequentially, in a random order, to avoid label oscillations in, e.g., bipartite or star-like graphs. Still, this severely hampers the stability of the approach [23, 26], since nodes that are updated at the beginning of some iteration gain higher propagation strength than those that are considered last.

Balancers  $b$  [26] decrease (increase) the strength of the nodes that are considered first (last), which counteracts for the introduced randomness and stabilizes the algorithm. Let  $t_i \in (0, 1]$  be a normalized index of node  $v_i$  in some random order. Assuming linearity, we can model node balancers simply as  $b_i = t_i$ , however, the use of a sigmoid curve allows for some further control [11] (see Alg. 1, line 7). The latter in fact introduces parameters  $\lambda$  and  $\eta \geq 0$ , where  $\lambda$  is fixed to 0.5 [26], while  $\eta$  represents the balance between the algorithm stability and complexity (see Section 4).

As label propagation in Eq. (4) can discover only densely connected groups of nodes, it is limited to communities. Nevertheless, the same principle can be extended to modules of nodes [4]. Rather than propagating the labels between the neighboring nodes, labels are propagated between the nodes at distance two (i.e., through common neighbors). Since nodes in modules share many neighbors, similarly as before, they form a consensus on some particular label  $g$ . Thus, when the propagation process unfolds,  $\{\mathcal{G}\}$  contains modules that are well depicted in the structure of the network [4, 11].

$$g_i = \operatorname{argmax}_g \left( \nu_g \cdot \overbrace{\sum_{v_j \in \Gamma_i} b_j f_j \cdot \delta(g_j, g)}^{\text{Community detection}} + (1 - \nu_g) \cdot \overbrace{\sum_{\substack{v_j \in \Gamma_i \\ v_k \in \Gamma_j \setminus \Gamma_i}} \frac{b_k f'_k}{k_j} \cdot \delta(g_k, g)}^{\text{Module detection}} \right) \quad (5)$$

The algorithm for detection of modules is shown in the right-hand side of Eq. (5). Preferences  $f'$  correspond to defensive propagation as above, while  $k_j$  in denominator makes the sum proportional to  $k_i$ . Otherwise, Eq. (5) represents a general group detection algorithm denoted General Propagation Algorithm [4]

---

**Algorithm 1** General Propagation Algorithm (*GPA*)

---

**Input:** Graph  $G(V, L)$  and parameters  $\lambda, \eta$  ( $\nu$ )

**Output:** Group partition  $\{\mathcal{G}\}$  (i.e., labels  $g$ )

```

1: for  $v_i \in V$  do
2:    $g_i \leftarrow i$  {Label initialization.}
3:    $f_i, f'_i \leftarrow 1/n$ 
4: end for
5: while not CONVERGENCE do
6:   for  $v_i \in \text{SHUFFLE}(V)$  do
7:      $b_i \leftarrow 1/(1 + e^{-\eta(t_i - \lambda)})$ 
8:      $g_i \leftarrow \text{Eq. (5)}$  {Label propagation.}
9:      $f_i \leftarrow \sum_{v_j \in \Gamma_i} f_j \cdot \delta(g_j, g_i) / \sum_{v_k \in \Gamma_j} \delta(g_k, g_i)$ 
10:     $f'_i \leftarrow \sum_{\substack{v_j \in \Gamma_i \\ v_k \in \Gamma_j \setminus \Gamma_i}} f'_k \cdot \delta(g_k, g_i) / \sum_{\substack{v_j \in \Gamma_k \\ v_l \in \Gamma_j \setminus \Gamma_k}} \delta(g_l, g_i)$ 
11:   end for
12: end while
13: return  $\{g_i\}$ 

```

---

(*GPA*), where intrinsic parameters  $\nu \in [0, 1]$  represent the adopted network modeling. Setting  $\nu_g = 1$  for all  $g$  is identical to community detection approach in Eq. (4), whereas, for all  $\nu_g$  equal to zero, the algorithm can reveal only modules. When  $\nu_g = 0.5$ , identified groups are based on community and module links.

Group detection framework [4, 11] in Alg. 1 is taken as the basis for the algorithm proposed in this paper. It ought, however, to be mentioned that one could else adopt an arbitrary group detection approach.

### 3.2. Hierarchical Propagation Algorithm

Group detection framework in Alg. 1 can reliably discover communities and also modules, when they are well defined in the network structure (see Section 4). Still, group parameters  $\nu$  that represent the adopted network modeling strategy have to be set accordingly. Approach in [4] estimates values of  $\nu_g$  during the propagation process by measuring the conductance [27] of each group  $g$ . Similarly, approach in [11] is based on a the standard clustering coefficient  $c$  [19]. However, due to an unsupervised nature of these strategies, both approaches suffer from the drifting effect, especially in larger networks.

We here propose a simpler strategy that fixes parameters  $\nu$  apriori during the label initialization step (Alg. 1, line 2). Recall that according to Section 2, communities appear in dense parts of real-world networks, when the density is measured by the degree-corrected clustering coefficient  $d$  [21]. On the contrary, clear modules appear in sparse parts with lower  $d$ . Thus, when label  $g_i$  of node  $v_i$  is initialized,

$$\nu_{g_i} = \begin{cases} 1 & \text{if } d_i \geq p \ (D \geq p), \\ 0 & \text{if } d_i < p \ (D < p), \\ 0.5 & \text{else,} \end{cases} \quad \begin{array}{l} (6a) \\ (6b) \\ (6c) \end{array}$$

where  $p$  is the expected network clustering in a random graph with the same degree sequence [20].<sup>1</sup>

$$p = \frac{(\sum_i k_i^2 - \langle k \rangle n)^2}{\langle k \rangle^3 n^3} \quad (7)$$

Since most real-world networks are dense and small-world with  $D \gg p$ , Eq. (6a) reveals communities in dense regions of the networks. On the other hand, Eq. (6b) properly models, e.g., two-mode networks with  $D = 0$ , where only modules are expected. However, as modules can also appear in denser regions with  $d > p$

---

<sup>1</sup> $p$  is derived for an alternative, but similar, definition of network clustering  $C$ .

---

**Algorithm 2** Hierarchical Propagation Algorithm (*HPA*)

---

**Input:** Graph  $G(V, L)$ 
**Output:** Group hierarchy  $\mathcal{H}$ 

```

1:  $\mathcal{H} \leftarrow \text{HIERARCHY}(G)$  {Hierarchy discovery.}
2: if  $|G(\mathcal{H})| > 1$  then
3:    $\mathcal{H} \leftarrow \mathcal{H} \cup \text{HPA}(G(\mathcal{H}))$  {Group agglomeration.}
4: end if
5: return  $\mathcal{H}$ 
6: procedure  $\text{HIERARCHY}(G)$ 
7:    $\{\mathcal{G}_i\} \leftarrow \text{PROPAGATION}(G)$  {Group detection.}
8:   if  $|\{\mathcal{G}_i\}| = 1$  then
9:     return  $\mathcal{H}(\mathcal{G})$ 
10:  end if
11:   $\mathcal{H} \leftarrow \{\}$ 
12:  for all  $\{\mathcal{G}_i\}$  do
13:     $\mathcal{H}_{\mathcal{G}_i} \leftarrow \text{HIERARCHY}(G(\mathcal{G}_i))$  {Group refinement.}
14:    if  $\mathcal{L}(\mathcal{H}_{\mathcal{G}_i}) > \mathcal{L}(\mathcal{H}(\mathcal{G}_i)) \wedge G(\mathcal{G}_i)$  connected then
15:       $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_{\mathcal{G}_i}$ 
16:    else
17:       $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}(\mathcal{G}_i)$ 
18:    end if
19:  end for
20:  return  $\mathcal{H}$ 
21: end procedure

```

---

(see Section 2.2), the above strategy apparently ignores them. Nevertheless, the approach we propose below intentionally first discovers dependent modules (e.g., bipartite configurations) as merged into communities, when their presence is not immediately apparent from the structure of the network (with respect to  $d$ ).

We introduce a group refinement procedure for the propagation framework in Alg. 1. Let  $\{\mathcal{G}\}$  be an initial group partition revealed by the algorithm. Then, for each group  $\mathcal{G}$ ,  $|\mathcal{G}| \geq 3$ , the algorithm is further applied to a network induced by the nodes in  $\mathcal{G}$  denoted  $G(\mathcal{G})$ . As this refinements proceed recursively, an entire sub-hierarchy of groups is revealed for each group  $\mathcal{G}$ . Similarly, we reveal a super-hierarchy by applying the algorithm to a network induced by the initial groups  $\{\mathcal{G}\}$ . Here nodes in fact represent groups that are linked, when a link also exists in the original network. Final result of such hierarchical refinements and agglomerations is a complete hierarchy of groups  $\mathcal{H}$  (see Fig. 3b).

The presented approach reveals group hierarchy in a top-down and bottom-up fashion simultaneously. Whereas hierarchical investigation is somewhat common in community detection [2, 28], top-down group refinement has not yet been used particularly for the discovery of modules (for top-down community discovery see, e.g., [2, 29]). However, since modules in real-world networks are often not immediately apparent from the network structure (see Section 4.3), group refinement step is in fact crucial. An adequate group refinement procedure is thus, together with an improved network modeling strategy in Eq. (6), the main novelty of the proposed approach denoted Hierarchical Propagation Algorithm (*HPA*).

Details of the algorithm are given in Alg. 2. Observe that each revealed hierarchy  $\mathcal{H}_{\mathcal{G}}$  is compared against an alternative with a single group  $\mathcal{G}$  denoted  $\mathcal{H}(\mathcal{G})$  (Alg. 2, line 14). The hypothesis is tested using the likelihood  $\mathcal{L} \in (0, 1]$  of the hierarchies given the network observed [30]. Hence,

$$\mathcal{L}(\mathcal{H}) = \prod_{\mathcal{I}_i \in \mathcal{H}} \theta_i^{m_i} \cdot (1 - \theta_i)^{M_i - m_i}, \quad (8)$$

where  $\mathcal{I}_i$  are inner nodes of  $\mathcal{H}$  representing different groups of nodes  $\mathcal{G}_i$ ,  $m_i$  is the number of links between the groups represented by the sub-hierarchy rooted at  $\mathcal{I}_i$ ,  $M_i$  is the number of all possible links and  $\theta_i = m_i/M_i$

are the maximum likelihood estimators for  $\mathcal{H}$  [31].  $\log \mathcal{L}$  of a hierarchy with a single level is in fact the entropy of a corresponding blockmodel [32], while in Section 5.3 we report  $-\log \mathcal{L}$ , where smaller is better.

*HPA* algorithm reveals an entire group hierarchy  $\mathcal{H}$ , thus, when only a partition is required, we report the groups represented by the bottom-most inner nodes in  $\mathcal{H}$  (no group agglomerations are needed).

#### 4. Analysis of the algorithm

The proposed *HPA* algorithm is rigorously analyzed and compared against different alternatives.

##### 4.1. Complexity, stability and validity

We first analyze the behavior of the proposed algorithm with respect to the only parameter  $\eta$  corresponding to balanced propagation (see Section 3.1). Table 1 shows pairwise Normalized Variation of Information [33] (*NVI*) of the revealed group partitions for two real-world networks (see Table 3). *NVI* is a distance in the space of partitions, while lower values represent better correlation,  $NVI \in [0, 1]$ . Besides *HPA* algorithm, we also include the basic label propagation in Eq. (3) (*LPA* algorithm) and a general group detection approach in Eq. (5) (*GPA* algorithm). Note that *HPA* and *GPA* algorithms differ merely in the hierarchical group refinement procedure introduced in Section 3.2.

Table 1: Mean pairwise *NVI* of the revealed group partitions (and the number of algorithm iterations) over 100 runs.

Network	Algorithm	$\eta = 0$	$\eta = 0.25$	$\eta = 2$
<i>football</i>	<i>LPA</i>	0.099 (3.6)	-	-
	<i>GPA</i>	0.090 (3.6)	0.078 (3.9)	<b>0.060</b> (7.1)
	<i>HPA</i>	0.082 (5.6)	0.073 (5.8)	<b>0.065</b> (9.2)
<i>women</i>	<i>LPA</i>	0.260 (4.3)	-	-
	<i>GPA</i>	0.092 (3.1)	0.086 (3.2)	<b>0.064</b> (3.8)
	<i>HPA</i>	0.094 (3.1)	0.083 (3.2)	<b>0.061</b> (4.5)

While setting  $\eta = 0$  is identical to the standard propagation, increasing parameter  $\eta$  improves the stability of all algorithms and, consequently, also the robustness of the revealed group structure (see Table 1). However, the number of iterations needed for the propagation to converge also increases. Parameter  $\eta$  thus controls the balance between the algorithm’s stability and complexity. For the comparison in Section 5 we set  $\eta = 2$ , and  $\eta = 0.25$  for larger real-world networks, while  $\eta = 0$  for the analysis below.

We further analyze the complexity of the proposed approach. Each iteration of *LPA* algorithm for community detection takes  $\mathcal{O}(m)$ , while the number of iterations grows very slowly as, e.g.,  $\mathcal{O}(\log m)$  [25]. Algorithm complexity was also estimated to  $\mathcal{O}(m^{1.23})$  [24]. On the other hand, the complexity of each iteration of the general group detection algorithms *GPA* and *HPA* is obviously  $\mathcal{O}(\langle k \rangle m)$ .

Fig. 1a shows the number of iterations for different algorithms applied to a realistic graph model [34], where the parameter  $L$  in the legend refers to the number of nontrivial levels in the revealed group hierarchy. Although general group detection and hierarchical investigation indeed increase complexity of the propagation, the number of iterations still appears to grow no faster than  $\mathcal{O}(\log m)$  (mind the scales). There is a super-logarithmic increase in the case of *HPA* algorithm, but analysis on real-world networks in Section 5.3 reveals that  $L$  is commonly only slightly larger than in the case of a comparable community detection approach (i.e., only one or two refinements steps are needed). Thus, the complexity of the proposed *HPA* algorithm can be estimated to  $\mathcal{O}(\langle k \rangle m \log m)$ , which should scale to networks with a million links.

Last, we also validate *HPA* algorithm on Erdős-Rényi [35] random graphs that presumably contain no characteristic groups of nodes (Fig. 1b). When the network size exceeds a certain threshold, the algorithm reveals no group structure (i.e., classifies all nodes into a single group). Behavior of the general group detection approaches else differs from *LPA* algorithm for communities only, nevertheless, the proposed approach does not suffer from the same problems like, e.g., modularity [36] optimization [37] exemplified by *GMO* and *LUV* algorithms (see Section 5.1).

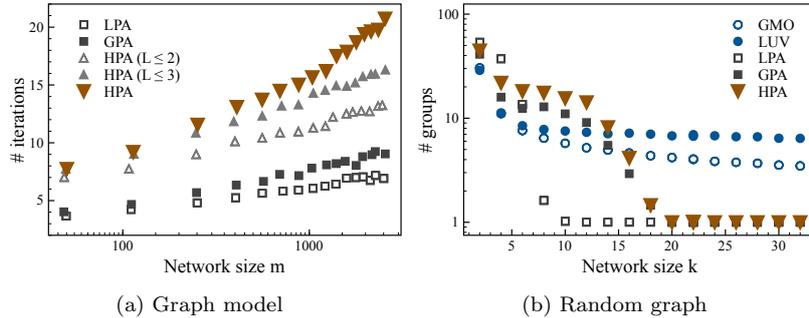


Figure 1: Mean number of (a) algorithm iterations and (b) revealed groups over 100 graph realizations.

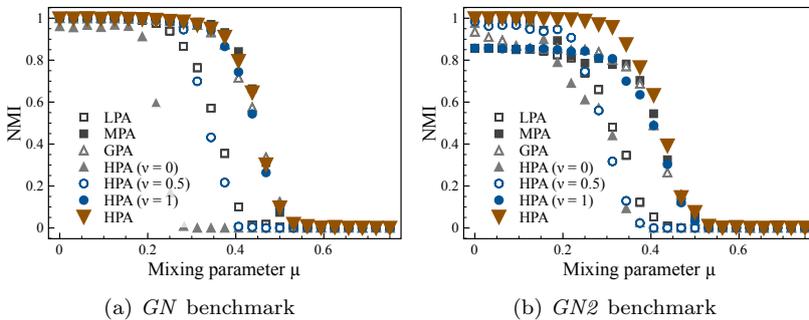


Figure 2: Mean  $NMI$  for group detection task over 100 graph realizations.

#### 4.2. Hierarchical group detection

Below we compare *HPA* algorithm against several alternatives on two synthetic benchmark graphs (see Table 2) with planted partitions of four communities (Fig. 2a) and two communities and two modules (Fig. 2b). Distribution of links is controlled by a mixing parameter  $\mu \in [0, 1]$ , where lower values correspond to a clearer group structure. Results are reported in the form of Normalized Mutual Information [3] ( $NMI$ ), which has become a *de facto* standard in the literature,  $NMI \in [0, 1]$  (higher is better).

Observe that *HPA* algorithm limited to module detection (i.e.,  $\nu = 0$ ) cannot accurately reveal communities for larger  $\mu$  (see Fig. 2a), while community detection approach (i.e.,  $\nu = 1$ ) completely fails at module detection (see Fig. 2b). On the other hand, approach with  $\nu = 0.5$  can detect general groups of nodes but only for smaller  $\mu$ . Only the proposed *HPA* algorithm can accurately reveal both communities and modules, and clearly outperforms all other alternatives including the best existing network group modeling strategy [11] (i.e., *MPA* algorithm). Note also the difference between proposed approach and *GPA* algorithm in Fig. 2b, which corresponds to the increase of performance due to an adequate hierarchical group refinement. Particularly, both approaches detect three dense groups of nodes (i.e., communities), whereas only *HPA* algorithm further refines one of these into two modules that were in fact planted into the network.

#### 4.3. Football social network

Fig. 3 further demonstrates the benefits of a general group detection approach like *HPA* algorithm. We apply the algorithm to a famous *football* social network that represents matches played among US college football teams in the 2000 season [2] (see Table 3). The latter has been of considerable interest in the community detection literature in the past, since the network reveals clear communities that coincide with the division into conferences. Revealed group hierarchy in Fig. 3b indeed contains communities on higher levels, however, several of these are then refined into well defined configurations of modules on lower levels. For instance, group of nodes at the top of Fig. 3a is in fact a complete multi-partite graph on ten nodes.

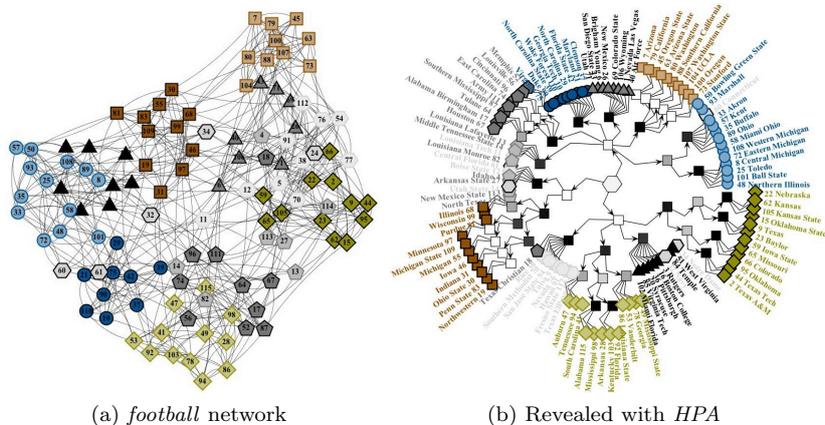


Figure 3: Group hierarchy of *football* network with  $\log \mathcal{L} = -970.2$ , where node shapes correspond to a known sociological division into groups. (Shades of the inner nodes of the hierarchy are proportional to probabilities  $\theta$ .)

Whether the particular group hierarchy would be of any interest in practical applications remains unclear, still, most of the groups present would remain overlooked under the standard community framework.

## 5. Comparison of the state-of-the-art

The proposed *HPA* algorithm is compared against the current state-of-the-art approaches in network group detection. In the following, we first present the experimental framework.

### 5.1. Experimental framework

Different approaches are first compared on synthetic benchmark graphs with planted group partition (Table 2). We adopt classical community detection benchmark that contains four equally-sized communities [2] (*GN* benchmark) and two variations of a more realistic benchmark graphs with scale-free degree and community size distributions [38] (*LFR* benchmarks). For a general group detection task, we adopt a generalization of the *GN* benchmark with two communities and two modules that are combined into a bipartite configuration [4] (*GN2* benchmark), and also two benchmarks with heterogeneous group size distribution and planted communities and bipartite or tripartite configurations (*SB2* and *SB3* benchmarks).

Table 2: Synthetic benchmark graphs used in the comparison of the state-of-the-art (“Groups” corresponds to either the number of communities and modules or community sizes).

Graph	Description	$n$	$\approx m$	Groups
<i>GN</i>	Classical community benchmark [2]	128	1024	4 – 0
<i>LFR</i>	Reliable community benchmark with power-law group size distribution [38]	1000	9800	[10, 50] [20, 100]
<i>GN2</i>	Generalization of <i>GN</i> benchmark [4]	128	1024	2 – 2
<i>SB2</i>	Partly bipartite graph with communities	112	936	3 – 4
<i>SB3</i>	Partly tripartite graph with communities	112	1448	3 – 3

Besides synthetic benchmark graphs, we also compare the approaches on ten real-world networks (Table 3). These were selected thus to include most types of networks that are commonly analyzed in the literature (e.g., social or information networks), whereas detailed description is omitted here. Note that, due to a large number of experimental settings in the paper, the networks are only of moderate size.

Table 3: Real-world networks used in the comparison of the state-of-the-art (*women* and *corporate* are two-mode networks).

Network	Description	$n$	$m$
<i>football</i>	US college football league [2]	115	613
<i>science</i>	Network science co-authorships [39]	1589	2742
<i>books</i>	Co-purchased political books [40]	105	441
<i>europe</i>	European highway network [26]	1039	1305
<i>jung</i>	JUNG network analysis library [41]	317	719
<i>compute</i>	Colt scientific computing library [41]	520	3691
<i>java</i>	java namespace of Java library [41]	1595	5287
<i>elegans</i>	C. elegans metabolic pathways [42]	453	2025
<i>women</i>	Southern women affiliations [43]	32	89
<i>corporate</i>	Scottish corporate interlocks [44]	217	348

*HPA* algorithm is compared against twelve other state-of-the-art approaches for group detection. For community detection task, we consider the following: greedy optimization of modularity [28] (*GMO* algorithm), multi-stage modularity optimization or Louvain method [45] (*LUV* algorithm), sequential clique percolation approach [46] (*SCP* algorithm), Markov clustering algorithm [47] (*MCL* algorithm), structural compression approach or Infomod [48] (*IMD* algorithm), random walk-based compression known as Infomap [49] (*IMP* algorithm) and the basic label propagation algorithm [23] (*LPA* algorithm). (*SCP* algorithm reveals overlapping groups, thus, each node in multiple groups is classified into a random one.)

For general group detection task, we adopt the following approaches: symmetric nonnegative matrix factorization [50] (*NMF* algorithm),  $k$ -means data clustering [51] based on [52] (*KMN* algorithm), mixture models (i.e., stochastic blockmodel) estimated by expectation-maximization algorithm [12] (*EMM* algorithm), mixture models with degree corrections [53] (*DMM* algorithm), model-based propagation algorithm [11] (*MPA* algorithm), structural compression approach [48] (*IMD* algorithm) and the best community detection algorithm above [49] (*IMP* algorithm). Note that *NMF*, *KMN*, *EMM* and *DMM* algorithms demand the number of groups apriori, which is a big disadvantage in practice. (Due to stability issues, *NMF* and *DMM* algorithms are applied to each network ten times and the best revealed group partition is reported.)

Due to limited resources, we do not include some otherwise very prominent approaches like [54, 55].

## 5.2. Comparison on synthetic graphs

Fig. 4 presents the results of the comparison on group detection task for the case of various synthetic benchmark graphs with planted group partition (see Table 2). Benchmark graphs in the top row contain only communities, whereas those in bottom row contain also different configurations of modules. Group structure is again controlled by a mixing parameter  $\mu$ , while results are reported as *NMI* (Section 4.2).

*IMP* algorithm outperforms all other approaches in community detection (Fig. 4, top), while the proposed *HPA* algorithm is comparable to the second-best *LUV* approach. Results are consistent with an empirical comparison of a larger number of approaches in [56], where none could improve on *IMP* algorithm (*LUV* algorithm performed similarly as above). *HPA* algorithm can thus be considered at least comparable to the current state-of-the-art on community detection task.

Note, however, that none of the above community-specialized approaches can discover general groups of nodes. In fact, only the proposed *HPA* algorithm can accurately detect the planted modules in the bottom row of Fig. 4 and notably improves on the existing *MPA* algorithm. Although not immediately evident, data clustering approach (i.e., *KMN* algorithm) performs surprisingly well on module detection task, but it fails in community detection, due to a loss of information about the network structure during transformation. Otherwise, only mixture models (i.e., *EMM* algorithm) that were very popular in the literature lately can also detect different groups of nodes planted in these benchmark graphs. Still, observe that the performance significantly decreases in the case of groups with heterogeneous size distribution (e.g., Fig. 4f), whereas the improved model that corrects for node degrees (i.e., *DMM* algorithm) fails on these benchmarks, due

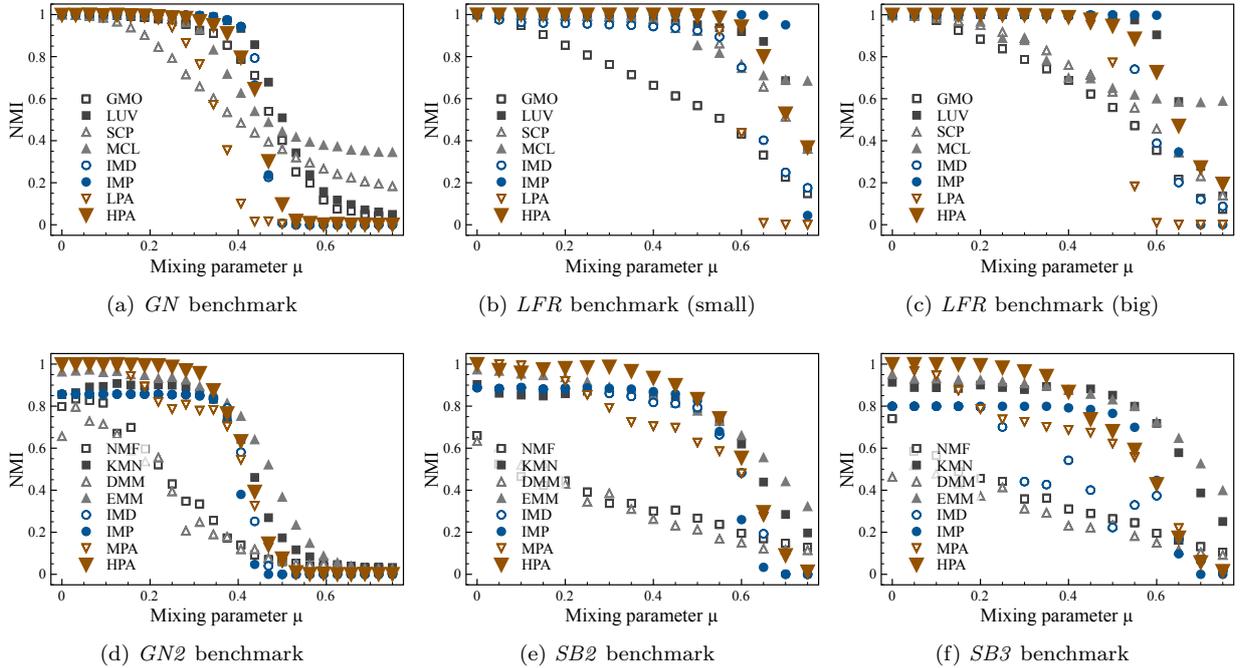


Figure 4: Mean *NMI* for group detection task over 100 graph realizations (10 for *MCL* and *DMM* algorithms).

to increased complexity. Other approaches also do not perform well (e.g., *NMF* algorithm). It should be stressed that despite relatively good performance of *KMN* and *EMM* algorithms, both approaches require the number of groups a priori, which seriously limits their applicability in practice [53]. The latter does not hold for the proposed *HPA* algorithm.

### 5.3. Comparison on real-world networks

In the following, we present results of the comparison of the state-of-the-art approaches on real-world networks. We consider network group detection, hierarchy discovery and link prediction tasks.

#### 5.3.1. Network group detection

The approaches are first compared on group detection task on two real-world networks with a known sociological division into groups, i.e., *football* social network with twelve communities introduced in Section 4.3 and a two-mode *women* affiliation network with three modules [43] (see Table 3). Results are reported in the form of *NMI* (Section 4.2) and Adjusted Rand Index [57] (*ARI*), where higher is better,  $ARI \in [0, 1]$ .

Table 4: Mean *NMI* and *ARI* for sociological group detection task over 100 runs.

Network	<i>LPA</i>	<i>LUV</i>	<i>IMP</i>	<i>KMN</i>	<i>EMM</i>	<i>HPA</i>
<i>football</i>	0.892	0.876	<b>0.922</b>	0.845	0.823	0.909
	0.796	0.771	<b>0.890</b>	0.698	0.683	0.850
<i>women</i>	0.184	0.309	0.417	0.677	0.827	<b>0.932</b>
	0.093	0.174	0.273	0.560	0.720	<b>0.936</b>

Results in Table 4 show similar performance as in the case of synthetic graphs in Section 5.2. *IMP* algorithm is the best approach for community detection (i.e., *football* network), while the proposed *HPA*

algorithm again performs well. In contrast to before, *HPA* algorithm clearly outperforms *LUV* algorithm (according to *ARI*). Furthermore, *HPA* algorithm is also a clear winner in module detection (i.e., *women* network), whereas *KMN* and *EMM* algorithms fail on these networks.

### 5.3.2. Network hierarchy discovery

We further compare the entire hierarchies of groups revealed with different state-of-the-art approaches. For a fair comparison, all approaches are extended with the hierarchical group agglomeration procedure proposed in *HPA* algorithm. Table 5 shows  $\log \mathcal{L}$ -s that correspond to posterior probabilities of the revealed hierarchies, where higher values are better,  $\log \mathcal{L} \leq 0$  (Section 3.2). Expectedly, group hierarchies discovered with *HPA* algorithm appear most likely, still, this is at least partly due to a larger search space. However, as claimed in Section 4.1, the number of nontrivial levels in the group hierarchies revealed with *HPA* algorithm is only slightly larger than for a comparable community detection approach (e.g., *LUV* algorithm).

Table 5: Peak  $-\log \mathcal{L}$  (and the number of nontrivial levels) of the group hierarchies revealed over 100 runs.

Network	<i>LUV</i>	<i>IMP</i>	<i>EMM</i>	<i>HPA</i>
<i>football</i>	1119.3 (2)	1066.7 (1)	1144.7 (2)	<b>1016.8 (3)</b>
<i>science</i>	2542.6 (3)	2106.3 (3)	-	<b>2041.4 (4)</b>
<i>europa</i>	5819.3 (2)	4410.2 (3)	-	<b>3981.0 (5)</b>
<i>jung</i>	2540.3 (2)	2302.9 (2)	-	<b>2259.9 (4)</b>
<i>javax</i>	13515.4 (2)	12438.3 (2)	-	<b>11512.2 (3)</b>
<i>women</i>	196.4 (2)	203.2 (1)	<b>163.6 (1)</b>	<b>163.6 (1)</b>

Note that the actual performance of approaches in practical applications cannot be reliably estimated from Table 5, Nevertheless, we stress that the revealed group structures differ considerably between the approaches. Fig. 5 shows the cumulative size distributions of the groups revealed in two real-world networks (see Table 3). These are power-law in the case of *HPA* algorithm, which is consistent with other analyses in the literature [28, 5], however, the distributions in Fig. 5a do not appear to be heavy-tailed in the case of a community detection approach (i.e., *IMP* algorithm). On the other hand, due to a resolution limit problem [37], optimization of modularity [36] clearly favors larger groups (i.e., *LUV* algorithm).

We also analyze the difference between the proposed *HPA* algorithm and a state-of-the-art community detection approach (i.e., *IMP* algorithm) on a *jung* software network [41] (see Table 3). Here nodes correspond to software classes, while the characteristic groups of nodes can be related to packages of the corresponding software library [41, 8]. Fig. 6 shows the revealed group hierarchies for only a part of *jung* network (to ease the comprehension). Both approaches reveal dense groups of nodes that correspond to graph implementations and parsers, I/O functionality and other (e.g., Fig. 6b). However, *HPA* algorithm also further partitions, e.g., graph parsers into implementations and meta classes (i.e., Fig. 6c, bottom).

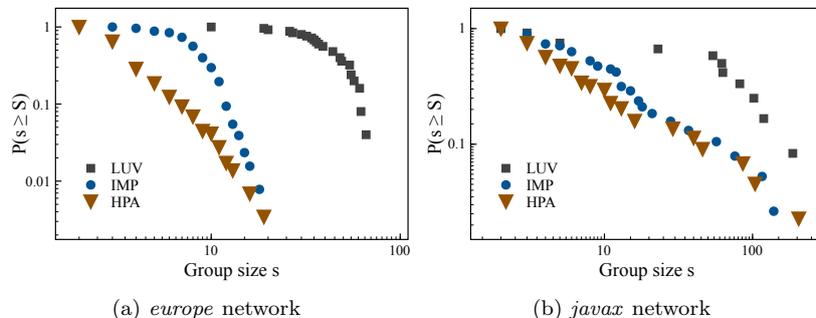


Figure 5: Cumulative size distributions of the group hierarchies with peak  $\log \mathcal{L}$  revealed over 100 runs.

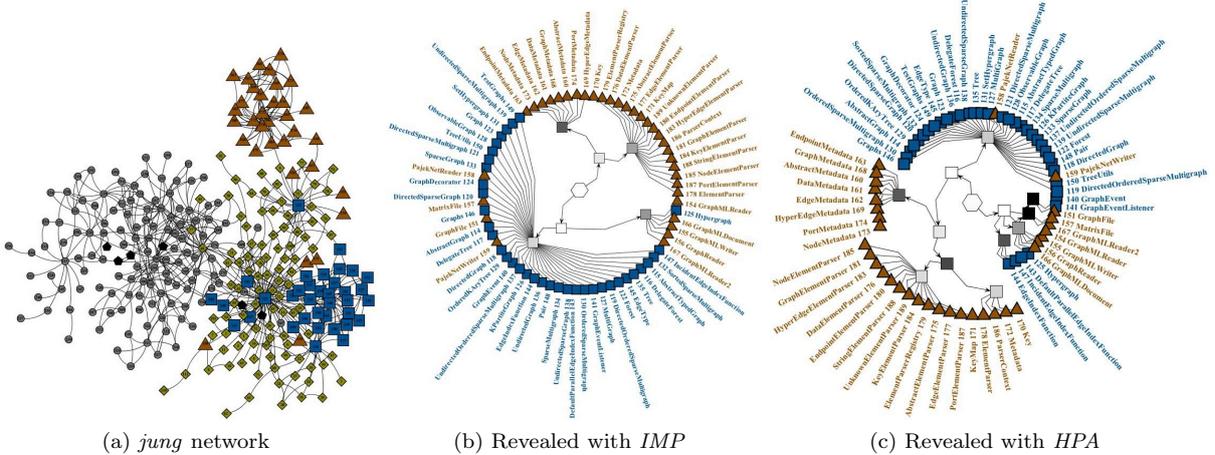


Figure 6: Group hierarchies of *jung* network with (b)  $\log \mathcal{L} = -500.9$  and (c)  $\log \mathcal{L} = -445.3$ , where node shapes correspond to high-level packages of JUNG network library, i.e., **graph** (squares), **io** (triangles), **algorithms** (diamonds), **visualization** (circles) and other (pentagons). (Shades of the inner nodes of the hierarchy are proportional to probabilities  $\theta$ .)

### 5.3.3. Network link prediction

Last, we compare *HPA* algorithm with the best community detection approach above (i.e., *IMP* algorithm) on link prediction task. We consider eight real-world networks from Table 3, where 5% of the links have been removed to represent positive examples (independently, for each network realization). Approximately the same number of disconnected pairs of nodes were selected for negative examples. Next, the algorithms were applied to each realization of the reduced network, revealing a group hierarchy  $\mathcal{H}$ . The probability of a link between a pair of nodes was predicted as  $\theta_i$ , where  $\mathcal{I}_i \in \mathcal{H}$  is the root of the smallest sub-hierarchy including both of the nodes (see Section 3.2). Results are shown in the form of Area Under the ROC Curve [58] (*AUC*), where higher values are better,  $AUC \in [0, 1]$ .

Table 6: Mean *AUC* for link prediction task over 100 network realizations (see text for details).

Network	$ \Gamma_i  \cdot  \Gamma_j $	$ \Gamma_i \cap \Gamma_j $	<i>IMP</i>	<i>HPA</i>
<i>football</i>	0.222	0.817	<b>0.804</b>	<b>0.799</b>
<i>science</i>	0.640	0.956	<b>0.934</b>	0.880
<i>books</i>	0.646	0.862	<b>0.763</b>	<b>0.762</b>
<i>europe</i>	0.356	0.530	<b>0.745</b>	0.700
<i>compute</i>	0.779	0.826	0.724	<b>0.766</b>
<i>elegans</i>	0.812	0.920	0.631	<b>0.641</b>
<i>women</i>	0.564	0.290	0.578	<b>0.699</b>
<i>corporate</i>	0.456	0.481	0.649	<b>0.748</b>

General group detection does not appear to be beneficiary for link prediction in the case of dense social networks like *science* author collaboration network and, surprisingly, very sparse *europe* road network (Table 6). On the other hand, *HPA* algorithm most accurately predicts the underlying network structure in biological, software and two-mode social networks. Thus, the proposed group hierarchy discovery could indeed be superior to the current state-of-the-art in practice. Merely for reference, Table 6 also shows two standard link prediction techniques based on preferential attachment graph model [59] (i.e.,  $|\Gamma_i| \cdot |\Gamma_j|$ ) and node equivalence [17] (i.e.,  $|\Gamma_i \cap \Gamma_j|$ ).

## 6. Conclusions and future work

The paper proposes a propagation-based general group detection algorithm for large real-world networks. The algorithm, in contrast to many other approaches, requires no apriori knowledge about the structure of the network (e.g., number of groups), while its computational complexity is near ideal. Moreover, rigorous analysis on group detection, hierarchy discovery and link prediction tasks reveals that the proposed approach is at least comparable to the current state-of-the-art. The main novelty of the paper is else a simple hierarchical refinement procedure that enables straightforward discovery of different types of groups (i.e., communities and modules), while the adopted methodology could be easily transferred to other approaches.

The paper also includes a rather extensive comparison of a larger number of group detection approaches on different synthetic benchmark graphs and real-world networks. Results give some interesting observations for future work. Most notably, while different approaches can accurately solve the community detection problem [45, 49], there is an absence of reliable algorithms for other groups of nodes. However, the approach based on data clustering [52] performs surprisingly well on module detection task. Thus, the approach could be combined with some community detection algorithm, and the group refinement procedure proposed in this paper, into a state-of-the-art hybrid approach. Next, mixture models [12, 53] that have been very popular in recent literature are a reliable approach for group detection only in networks of moderate size, mainly due to a simple optimization procedure. Also, these require the number of groups to be known apriori, which remains to be an open problem in network science [53]. On the other hand, methods based on dynamical processes like [49, 24], and the algorithm proposed in the paper, appear to be the most reliable approaches for group detection without the above weaknesses.

The comparison in the paper does not include some otherwise very prominent group detection techniques [54, 55] that will be included in future work. Also, the proposed group refinement procedure may fail to recognize some less clear group configurations, which remains to be thoroughly investigated.

## Acknowledgments

The work has been supported by the Slovene Research Agency *ARRS* within the research program no. P2-0359.

## References

- [1] M. E. J. Newman, Communities, modules and large-scale structure in networks, *Nat. Phys.* 8 (1) (2012) 25–31.
- [2] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, *P. Natl. Acad. Sci. USA* 99 (12) (2002) 7821–7826.
- [3] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, *J. Stat. Mech.* P09008.
- [4] L. Šubelj, M. Bajec, Ubiquitousness of link-density and link-pattern communities in real-world networks, *Eur. Phys. J. B* 85 (1) (2012) 32.
- [5] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature* 435 (7043) (2005) 814–818.
- [6] J. Ruan, W. Zhang, An efficient spectral algorithm for network community discovery and its applications to biological and social networks, in: *Proceedings of the IEEE International Conference on Data Mining, 2007*, pp. 643–648.
- [7] J. Leskovec, L. A. Adamic, B. A. Huberman, The dynamics of viral marketing, *ACM Trans. Web* 1 (1).
- [8] L. Šubelj, M. Bajec, Software systems through complex networks science: Review, analysis and applications, in: *Proceedings of the KDD Workshop on Software Mining, Beijing, China, 2012*, pp. 9–16.
- [9] P. Boldi, M. Rosa, M. Santini, S. Vigna, Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks, in: *Proceedings of the International World Wide Web Conference, Hyderabad, India, 2011*, pp. 587–596.
- [10] S. Fortunato, Community detection in graphs, *Phys. Rep.* 486 (3-5) (2010) 75–174.
- [11] L. Šubelj, M. Bajec, Generalized network community detection, in: *Proceedings of the ECML PKDD Workshop on Finding Patterns of Human Behaviors in Network and Mobility Data, Athens, Greece, 2011*, pp. 66–84.
- [12] M. E. J. Newman, E. A. Leicht, Mixture models and exploratory analysis in networks, *P. Natl. Acad. Sci. USA* 104 (23) (2007) 9564.
- [13] B. Long, X. Xu, Z. Zhang, P. Yu, Community learning by graph approximation, in: *Proceedings of the IEEE International Conference on Data Mining, Omaha, NE, USA, 2007*, pp. 232–241.
- [14] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, *P. Natl. Acad. Sci. USA* 101 (9) (2004) 2658–2663.

- [15] S. Pinkert, J. Schultz, J. Reichardt, Protein interaction networks: More than mere modules, *PLoS Comput. Biol.* 6 (1) (2010) e1000659.
- [16] J. Reichardt, D. R. White, Role models for complex networks, *Eur. Phys. J. B* 60 (2) (2007) 217–224.
- [17] F. Lorrain, H. C. White, Structural equivalence of individuals in social networks, *J. Math. Sociol.* 1 (1) (1971) 49–80.
- [18] M. G. Everett, S. P. Borgatti, Regular equivalence: General theory, *J. Math. Sociol.* 19 (1) (1994) 29–52.
- [19] D. J. Watts, S. H. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature* 393 (6684) (1998) 440–442.
- [20] M. E. J. Newman, S. H. Strogatz, D. J. Watts, Random graphs with arbitrary degree distributions and their applications, *Phys. Rev. E* 64 (2) (2001) 026118.
- [21] S. N. Soffer, A. Vázquez, Network clustering coefficient without degree-correlation biases, *Phys. Rev. E* 71 (5) (2005) 057101.
- [22] M. E. J. Newman, Mixing patterns in networks, *Phys. Rev. E* 67 (2) (2003) 026126.
- [23] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E* 76 (3) (2007) 036106.
- [24] L. Šubelj, M. Bajec, Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extraction, *Phys. Rev. E* 83 (3) (2011) 036103.
- [25] I. X. Y. Leung, P. Hui, P. Liò, J. Crowcroft, Towards real-time community detection in large networks, *Phys. Rev. E* 79 (6) (2009) 066107.
- [26] L. Šubelj, M. Bajec, Robust network community detection using balanced propagation, *Eur. Phys. J. B* 81 (3) (2011) 353–362.
- [27] B. Bollobás, *Modern Graph Theory*, Springer, Heidelberg, 1998.
- [28] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, *Phys. Rev. E* 70 (6) (2004) 066111.
- [29] M. E. J. Newman, Modularity and community structure in networks, *P. Natl. Acad. Sci. USA* 103 (23) (2006) 8577–8582.
- [30] A. Clauset, C. Moore, M. E. J. Newman, Structural inference of hierarchies in networks, in: *Proceedings of the ICML Workshop on Statistical Network Analysis*, Pittsburgh, PA, USA, 2006, pp. 1–13.
- [31] G. Casella, R. L. Berger, *Statistical Inference*, Duxbury Press, Belmont, 1990.
- [32] T. P. Peixoto, The entropy of stochastic blockmodel ensembles, *Phys. Rev. E* 85 (5) (2012) 056122.
- [33] B. Karrer, E. Levina, M. E. J. Newman, Robustness of community structure in networks, *Phys. Rev. E* 77 (4) (2008) 046119.
- [34] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, *ACM Trans. Knowl. Discov. Data* 1 (1) (2007) 1–41.
- [35] P. Erdős, A. Rényi, On random graphs i, *Publ. Math. Debrecen* 6 (1959) 290–297.
- [36] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Phys. Rev. E* 69 (2) (2004) 026113.
- [37] S. Fortunato, M. Barthélemy, Resolution limit in community detection, *P. Natl. Acad. Sci. USA* 104 (1) (2007) 36–41.
- [38] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, *Phys. Rev. E* 78 (4) (2008) 046110.
- [39] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E* 74 (3) (2006) 036104.
- [40] V. Krebs, *A network of co-purchased books about US politics* (2006).
- [41] L. Šubelj, M. Bajec, Community structure of complex software systems: Analysis and applications, *Physica A* 390 (16) (2011) 2968–2975.
- [42] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, A.-L. Barabási, The large-scale organization of metabolic networks, *Nature* 407 (6804) (2000) 651–654.
- [43] A. Davis, B. B. Gardner, M. R. Gardner, *Deep South*, Chicago University Press, Chicago, 1941.
- [44] J. Scott, M. Hughes, *The anatomy of Scottish capital: Scottish companies and Scottish capital, 1900-1979*, Croom Helm, 1980.
- [45] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *J. Stat. Mech.* P10008.
- [46] J. M. Kumpula, M. Kivelä, K. Kaski, J. Saramäki, Sequential algorithm for fast clique percolation, *Phys. Rev. E* 78 (2) (2008) 026109.
- [47] S. van Dongen, *Graph clustering by flow simulation*, PhD thesis, University of Utrecht (2000).
- [48] M. Rosvall, C. T. Bergstrom, An information-theoretic framework for resolving community structure in complex networks, *P. Natl. Acad. Sci. USA* 104 (18) (2007) 7327–7331.
- [49] M. Rosvall, C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, *P. Natl. Acad. Sci. USA* 105 (4) (2008) 1118–1123.
- [50] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix tri-factorizations for clustering, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, 2006, pp. 126–135.
- [51] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, USA, 1967, pp. 281–297.
- [52] C.-Y. Lin, J.-L. Koh, A. L. P. Chen, A better strategy of discovering link-pattern based communities by classical clustering methods, in: *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Hyderabad, India, 2010, pp. 56–67.
- [53] B. Karrer, M. E. J. Newman, Stochastic blockmodels and community structure in networks, *Phys. Rev. E* 83 (1) (2011)

016107.

- [54] K. Henderson, T. Eliassi-Rad, S. Papadimitriou, C. Faloutsos, HCDF: a hybrid community discovery framework, in: Proceedings of the SIAM International Conference on Data Mining, Columbus, OH, USA, 2010, pp. 754–765.
- [55] T. Nishikawa, A. E. Motter, Discovering network structure beyond communities, *Sci. Rep.* 1 (151) (2011) 1–7.
- [56] A. Lancichinetti, S. Fortunato, Community detection algorithms: A comparative analysis, *Phys. Rev. E* 80 (5) (2009) 056117.
- [57] L. Hubert, P. Arabie, Comparing partitions, *J. Classif.* 2 (1) (1985) 193–218.
- [58] J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve., *Radiology* 143 (1) (1982) 29–36.
- [59] A. L. Barabási, R. Albert, Emergence of scaling in random networks, *Science* 286 (5439) (1999) 509–512.