Leveraging disjoint communities for detecting overlapping community structure

J. Stat. Mech. (2015) P05017

(http://iopscience.iop.org/1742-5468/2015/5/P05017)

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Leveraging disjoint communities for detecting overlapping community structure

Tanmoy Chakraborty

Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur, West Bengal 721302, India
E-mail: its_tanmoy@cse.iitkgp.ernet.in

Received 17 March 2015
Accepted for publication 10 April 2015
Published 18 May 2015

Online at stacks.iop.org/JSTAT/2015/P05017
doi:10.1088/1742-5468/2015/05/P05017

Abstract. Network communities represent mesoscopic structure for understanding the organization of real-world networks, where nodes often belong to multiple communities and form overlapping community structure in the network. Due to non-triviality in finding the exact boundary of such overlapping communities, this problem has become challenging, and therefore huge effort has been devoted to detect overlapping communities from the network.

In this paper, we present PVOC (Permanence based Vertex-replication algorithm for Overlapping Community detection), a two-stage framework to detect overlapping community structure. We build on a novel observation that non-overlapping community structure detected by a standard disjoint community detection algorithm from a network has high resemblance with its actual overlapping community structure, except the overlapping part. Based on this observation, we posit that there is perhaps no need of building yet another overlapping community finding algorithm; but one can efficiently manipulate the output of any existing disjoint community finding algorithm to obtain the required overlapping structure. We propose a new *post-processing technique* that by combining with any existing disjoint community detection algorithm, can suitably process each vertex using a new vertex-based metric, called *permanence*, and thereby finds out overlapping candidates with their community memberships. Experimental results on both synthetic and large real-world networks show that PVOC significantly outperforms six state-of-the-art overlapping community detection algorithms in terms of high similarity of the output with the ground-truth structure. Thus our framework not only finds meaningful overlapping communities from the network, but also allows us to put an end to the constant effort of building yet another overlapping community detection algorithm.

Keywords: random graphs, networks, clustering techniques

Contents

1. Introduction

One of the most used aspects of social network analysis is to discover and display clusters and communities in networks—the dense sub-networks, where there are more links internally, than externally. It is easy for the common person to spot dense clusters of connection in a small network visualization. However, this is extremely difficult problem to detect such groups from large scale networks. There has been a constant effort since last one decade from the researchers of both computer science and physics domains to explore such community structure from networks after the pioneering effort of Girvan and Newman [1]. Today there are dozens of community detection algorithms that can detect the disjoint/non-overlapping community structure from the network using different heuristics and frameworks (see [2, 3] for the survey). However, in real-world scenario, it has been observed that a node can be a part of multiple communities, which has eventually led to the idea of overlapping/soft communities [4–7]. This problem is even more harder because of the exponential number of possible solutions. Therefore, a new direction of research has been started to detect the overlapping community structure from the network (see [8] for the survey).

The dichotomy between 'disjoint' and 'overlapping' community detection algorithms is unfortunate because it limits the application of each algorithm. If a network has overlapping communities, a 'disjoint' algorithm cannot find them; conversely, if communities are known to be disjoint, a 'disjoint' algorithm will generally perform better than an 'overlapping' algorithm. Therefore, to obtain the actual community structure, it is important to choose the right kind of algorithm. Note that the question of how to choose the right kind of algorithm is outside the scope of the present paper.

However, we hypothesize that there is perhaps no need to develop yet another overlapping community finding algorithm given the assumption that we have diverse and efficient disjoint community detection algorithms in hand. In this paper, we present a method to allow any 'disjoint' community detection algorithm to be used to detect overlapping community structure instead for finding another overlapping community detection algorithm. This means that a user wishing to find overlapping communities need no longer be forced to use one of the overlapping algorithms that exist, but can also choose from the many disjoint community finding algorithms. The proposed framework is called as PVOC (Permanence based Vertex-replication algorithm for Overlapping Community detection) which is a two-phase framework—in the first step, an efficient disjoint community detection algorithm is used to detect the non-overlapping community structure from the network; in the second step, each node in the disjoint communities is processed appropriately using a new vertex-based metric, called *permanence* [9], in order to measure the extent of belongingness of a vertex in its own community and its attached neighboring communities. If the membership of the vertex in its assigned community is similar to that in the neighboring community, we assign the vertex into the neighboring community, keeping its original community intact. Thus the post-processing step is the fundamental component in PVOC to find out overlapping vertices from the non-overlapping structure.

We compare our framework with six state-of-the-art overlapping community detection algorithms on both synthetic and large real-world networks (whose ground-truth

J. Stat. Mech. (2015) P05017

community structure is available). We observe that PVOC significantly outperforms other baseline algorithms in terms of high resemblance of the output with the ground-truth structure. Moreover, we show that even if it is scalable, it does not compromise the correctness of the output.

Our paper makes several unique contributions to the state-of-the-art in community detection. These include (i) analyzing the real-world community structure and observing that the disjoint communities are enough to be processed for discovering overlapping community structure, (ii) proposing a new framework by combining existing disjoint community detection algorithm along with the post-processing step, (iii) showing the accuracy of PVOC in terms of accurately discovering the ground-truth structure.

The organization of the paper is as follows. In the next section, we provide a brief overview of state-of-the-art approaches in overlapping community detection. Section 3 provides a brief description of the synthetic and real-world datasets. Following this, in section 4, we present a detailed results of our empirical observation followed by the description of our proposed framework. Section 5 describes the results of the experiments to detect overlapping communities and a comparative analysis with the baseline algorithms. The experiments in this paper use a combination of PVOC with two existing disjoint community detection algorithms, Louvain [10] and Infomap [11]. Finally, we conclude the paper in section 6 with some immediate future directions.

2. Related work

There has been a class of algorithms for network clustering, which allow nodes belonging to more than one community. Palla proposed 'CFinder' [12], the seminal and most popular method based on clique-percolation technique. However, due to the clique requirement and the sparseness of real networks, the communities discovered by CFinder are usually of low quality [13]. The idea of partitioning links instead of nodes to discover community structure has also been explored [14–17].

On the other hand, a set of algorithms utilized local expansion and optimization to detect overlapping communities. For instance, Baumes *et al* [18] proposed a two-step algorithm 'RankRemoval' using a local density function. LFM [19] expands communities from a random seed node to form a natural community until a fitness function is locally maximal. MONC [20] uses the modified fitness function of LFM which allows a single node to be considered a community by itself. OSLOM [5] tests the statistical significance of a cluster with respect to a global null model (i.e. the random graph generated by the configuration model) during community expansion. Chen *et al* [21] proposed selecting a node with maximal node strength based on two quantities—belonging degree and the modified modularity. EAGLE [6] and GCE [22] use the agglomerative framework to produce overlapping communities. COCD [23] first identifies cores and then remaining nodes are attached to cores with which they have maximum connections.

Few fuzzy community detection algorithms have been proposed that quantify the strength of association between all pairs of nodes and communities [24]. Nepusz *et al* [25] modeled the overlapping community detection as a nonlinear constrained optimization problem which can be solved by simulated annealing methods. Zhang *et al* [26] proposed an algorithm based on the spectral clustering framework. Due to the probabilistic nature, mixture models provide an appropriate framework for overlapping community detection

J. Stat. Mech. (2015) P05017

[27–30]. MOSES [31] uses a local optimization scheme in which the fitness function is defined based on the observed condition distribution. Zhang *et al* used nonnegative matrix factorization (NMF) to detect overlapping communities when the number of communities and the feature vectors are provided [32, 33]. Ding *et al* [34] employed the affinity propagation clustering algorithm for overlapping detection. Recently, the BIGCLAM [35] algorithm was also built on a NMF framework.

The label propagation algorithm has been extended to overlapping community detection by allowing a node to have multiple labels. In COPRA [36], each node updates its belonging coefficients by averaging the coefficients from all its neighbors at each time step in a synchronous fashion. SLPA [37, 38] spreads labels between nodes according to pairwise interaction rules. A game-theoretic framework is proposed in Chen *et al* [39] in which a community is associated with a Nash local equilibrium.

Beside these, CONGA [7] extends GN algorithm [40] by allowing a node to split into multiple copies. Zhang *et al* [41] proposed an iterative process that reinforces the network topology and propinquity that is interpreted as the probability of a pair of nodes belonging to the same community. István *et al* [42] proposed an approach focusing on centrality-based influence functions. Recently, Gopalan and Blei [43] proposed an algorithm that naturally interleaves subsampling from the network and updating an estimate of its communities. The reader can get more details in a nice survey paper by Xie *et al* [8].

3. Test suite of networks

3.1. Synthetic networks

It is necessary to have good benchmarks to both study the behavior of a proposed community detection algorithm and to compare the performance across various algorithms. In light of this requirement, Lancichinetti *et al* [44] introduced LFR[1] benchmark networks that take into account heterogeneity into degree and community size distributions of a network. These distributions are governed by power laws with exponents τ_1 and τ_2 respectively. To generate overlapping communities O_n, the fraction of overlapping nodes is specified and each node is assigned to O_m ($\geqslant$1) communities. LFR also provides a rich set of parameters to control the network topology, including the number of nodes n, the mixing parameter μ, the average degree $\bar{k}$, the maximum degree $k_{\max}$, the maximum community size $c_{\max}$, and the minimum community size $c_{\min}$. We vary these parameters depending on the experimental needs. Unless otherwise stated, LFR graph is generated with the following configuration: $\mu = 0.2$, $N = 10\,000$, $O_m = 4$, $O_n = 5\%$; other parameters being set to their default values. Results shown are the average of 100 runs.

3.2. Real-world networks with ground-truth communities

We use four real-world networks[2] proposed by Yang and Leskovec [35, 45] whose underlying ground-truth community structures are known a priori and whose properties are summarized in table 1. Figure 1 shows the distribution of the number of communities memberships of vertices for the real-world networks.

[1] http://sites.google.com/site/andrealancichinetti/files.
[2] http://snap.stanford.edu.

Table 1. Properties of the real-world networks used in this experiments. N: number of nodes, C: number of communities, S: average size of a community, $\bar{O}_m$: average number of community memberships per node.

Networks	N	E	C	S	$\bar{O}_m$
DBLP	317 080	1049 866	13 477	429.79	2.57
Amazon	334 863	925 872	151 037	99.86	14.83
Youtube	1134 890	2987 624	8385	9.75	10.26
Orkut	3072 441	117 185 083	6288 363	34.86	95.93

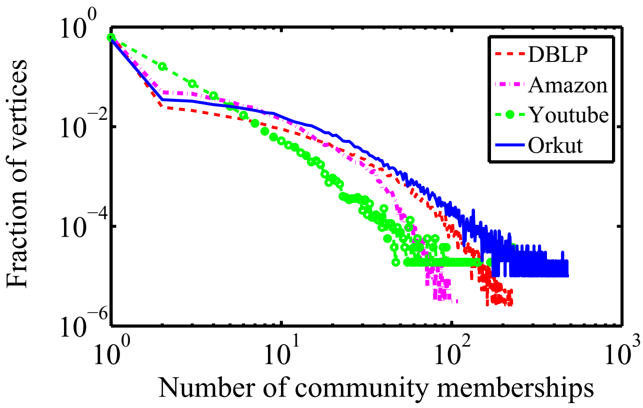


Figure 1. Distribution of the number of community memberships of vertices. X-axis shows the number of community memberships of vertices and y-axis shows the fraction of vertices with certain number of community memberships.

DBLP: It is a co-authorship network where nodes represent authors and edges connect nodes whose corresponding authors have co-authored in at least one paper. Since research communities stem around conferences or journals, the publication venues are used as ground-truth communities in DBLP.

Amazon: It is a Amazon product co-purchasing network where nodes represent products and edges connect commonly co-purchased products. Each product (i.e. node) belongs to one or more product categories. Each product category is used to define a ground-truth community.

Youtube: In the Youtube social network, users form friendship with each other and users can create groups where other users can join. Here, such user-defined groups are considered as ground-truth communities.

Orkut: Orkut is a free on-line social network where users form friendship with each other. Orkut also allows users to form a group where other members can then join. Here also such user-defined groups are considered as ground-truth communities.

4. Vertex-replication algorithm

Our proposed algorithm is motivated from an empirical study on the ground-truth community structure of both synthetic and real-world networks. In this section, we first describe the empirical observation and then illustrate a new algorithm that can detect

J. Stat. Mech. (2015) P05017

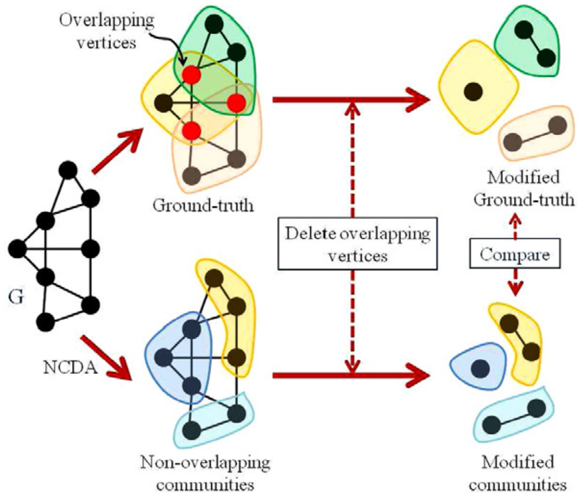


Figure 2. An illustrative example to show the procedure followed in our empirical study (NCDA: Non-overlapping Community Detection Algorithm).

overlapping communities from a network with the help of *any* standard disjoint community detection algorithm.

4.1. Empirical observation

We empirically study the structure of the ground-truth communities. We speculated that if we remove the vertices that are part of multiple communities from the ground-truth structure, the rest of the portion, i.e. the community structure composed of only non-overlapping vertices can be efficiently captured by the standard disjoint community detection algorithm. To verify this intuition, we take all the networks with their ground-truth communities and two standard disjoint community detection algorithms, namely Louvain [10] and Infomap [11, 46]. Then for each network, we run the following steps:

I We run each of these algorithms to obtain the disjoint community structure from the network.

II Since we know the ground-truth community structure of the network, we remove from the ground-truth those vertices (refer to set V_o) which belong to multiple communities.

III Similarly, we remove the constituent vertices of V_o from the disjoint community structure obtained from Step I. This step makes sure that the filtered ground-truth community structure and the filtered disjoint community structure obtained from the algorithm contain same set of vertices.

IV Then we compare two community structures obtained from Step II and Step III.

A schematic example of the above procedure is shown in figure 2. Figure 1 shows that in this process, we discard nearly 40% of the vertices (on an average) which belong to multiple communities for each network. In table 2, we also report the number of disjoint communities obtained from Louvain and Infomap algorithms for both synthetic

Table 2. Number of communities in the ground-truth structure and that obtained from Louvain and Infomap for LFR and real-world networks.

		Algorithms	
Networks	Ground-truth	Louvain	Infomap
LFR	582	468	501
DBLP	8493	7987	8145
Amazon	151 037	142 098	149 876
Youtube	8385	7967	7132
Orkut	288 363	284 980	286 791

Note: Here for the LFR network, we consider the following configuration: $N = 10\,000$, $\mu = 0.2$, $O_m = 4$, $O_n = 5\%$. The result of LFR is averaged over 100 runs.

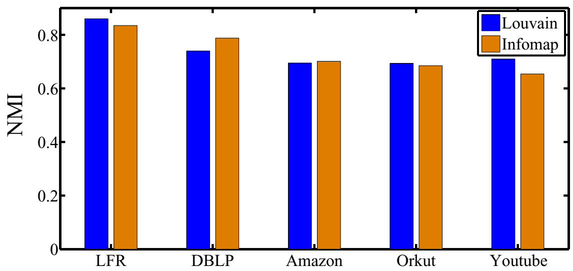


Figure 3. Similarity (in terms of NMI) of the community structure obtained from two disjoint community detection algorithms (Louvain and Infomap) with the ground-truth structures after excluding the overlapping vertices. Here for the LFR network, we consider the following configuration: $N = 10\,000$, $\mu = 0.2$, $O_m = 4$, $O_n = 5\%$.

and real-world networks and that present in the ground-truth structure. We use a standard validation metric, namely normalized mutual information (NMI) [47] to compare these two community structures. Figure 3 shows that the similarity is quite high for all the networks; this observation indeed corroborates our earlier speculation. Therefore, we hypothesize that a standard disjoint community detection algorithm might be able to find the overlapping communities with a suitable post-processing step. This means that a user wishing to find overlapping communities need no longer be forced to use any overlapping community finding algorithm, rather a disjoint community structure followed by a *post-processing step* might produce the expected overlapping community structure. In the rest of this section, we shall use this observation to design a suitable post-processing technique.

4.2. Permanence based vertex-replication algorithm

Through careful inspection mentioned above, we have found that a standard disjoint community detection algorithms are quite efficient to detect the non-overlapping part of the community structure. However there exist few vertices in the network, which are part of multiple communities. We intend to design an efficient algorithm that would be able to identify such overlapping vertices with their community memberships. For that, we use

J. Stat. Mech. (2015) P05017

a vertex-based metric, called *permanence*, which by virtue of its underlying formulation measures how intensely a vertex belongs to its community [9]. Below, we present a brief overview of the formulation of permanence.

4.2.1. Formulation of permanence. In an earlier paper [9], we showed that the extent of membership of a vertex to a community depends on the following two factors. (i) The first factor is *the distribution of external connections of the vertex to individual communities*. A vertex that has equal number of connections to all its external communities (e.g. a vertex with total 6 external connections with 2 to each of 3 neighboring communities) has equal 'pull' from each community whereas a vertex with more external connections to one particular community (e.g. a vertex with total 6 external connections with 1 connection each to two neighboring communities and 4 connections to the third neighboring community), will experience more 'pull' from that community due to large number of external connections to it. (ii) The second factor is *the density of its internal connections*. The internal connections of a community are generally considered together as a whole. However, how strongly a vertex is connected to its internal neighbors can differ. To measure this internal connectedness of a vertex, one can compute the clustering coefficient of the vertex with respect to its internal neighbors. The higher this internal clustering coefficient, the more tightly the vertex is connected to its community.

Combining these two factors together, we formulated permanence Perm(.) of a vertex v as follows:

$$\mathrm{Perm}(v) = \frac{I(v)}{E_{\max}(v)} \times \frac{1}{D(v)} - (1 - c_{in}(v)) \tag{1}$$

where $I(v)$ is the number of internal connections of v, $D(v)$ is the degree of v, $E_{\max}(v)$ is the maximum connections of v to a single external community and $c_{in}(v)$ is the clustering coefficient among the internal neighbors of v. An illustrative example is shown in figure 4.

For vertices that do not have any external connections, $\mathrm{Perm}(v)$ is considered to be equal to the internal clustering coefficient (i.e. $\mathrm{Perm}(v) = c_{in}(v)$). The maximum value of $\mathrm{Perm}(v)$ is 1 and is obtained when vertex v is an internal node and part of a clique. The lower bound of $\mathrm{Perm}(v)$ is close to -1. This is obtained when $I(v) \ll D(v)$, such that $\frac{I(v)}{D(v)E_{\max}(v)} \approx 0$ and $c_{in}(v) = 0$. Therefore for every vertex v, $-1 < \mathrm{Perm}(v) \leqslant 1$.

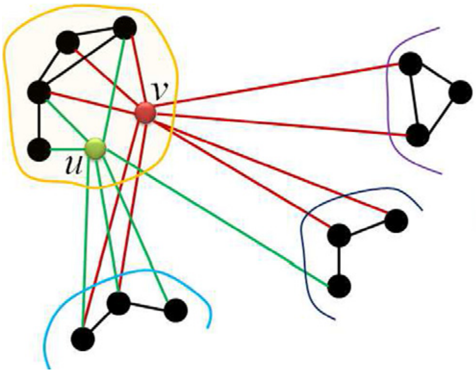
4.2.2. The PVOC algorithm. Since permanence can assign a score to each of the vertices, we can use it in our post-processing step to identify overlapping vertices from the detected disjoint community structure. Subsequently, we develop a new algorithm, called PVOC (Permanence based Vertex-replication algorithm for Overlapping Community detection) that can combine any existing disjoint community detection algorithm with the permanence based vertex-replication for detecting overlapping community structure of a network. Algorithm 1 presents the pseudo-code of PVOC.

Given undirected network $G(V, E)$ and a threshold θ, the algorithm works as follows:

I A standard disjoint community detection algorithm A_d is used to detect non-overlapping community structure NC from G.

II A set of vertices V_e are identified from NC such that each constituent vertex in V_e has at least one connection to any external community.

J. Stat. Mech. (2015) P05017



Vertex	D(.)	I(.)	E_{max}(.)	C_{in}(.)	Perm(.)
u	7	3	3	2/3	-0.191
v	9	3	2	1	0.167

Figure 4. Toy example depicting *permanence* of two vertices u and v. Even if vertex v has a large number of external connections than u, all these six external connections are distributed equally into three neighboring communities, resulting in the external pull proportional to 2; whereas u is attached with 4 external neighbors, three of them constitute in one community and the rest is attached with another community, resulting in the external pull proportional to 3. On the other hand, v is connected to 3 internal neighbors which are further completely connected among each other; whereas the neighbors of u are partially connected. This results in high internal pull of v as compared to u. These two notions of connectively are considered in the formulation of permanence.

III For each vertex v in V_e, we do the following steps:

(a) We calculate the sum of permanence of v and its neighbors in their assigned communities.

(b) We remove v from its own community and place it to each of its external communities separately. This assignment affects the permanence value of v and its immediate neighbors.

(c) For each external community C_n, we measure the current sum of permanence of v (in its new community) and its neighbors.

(d) If the absolute value of the difference of the permanence values obtained from Step III(a) and Step III(c) is less than θ, a replica of v is placed into the new community C_n, keeping v in its original community as well; otherwise v is assigned back to its original community. This step identifies overlapping nodes along with their memberships in different communities.

(e) The algorithm finally returns all the vertices with new community membership.

The threshold θ controls the extent to which one can relax the condition of replicating a vertex into multiple communities. We vary the threshold from 0 to 0.2 and observe that it produces maximum accuracy at 0.05 (see figure 6). Therefore, for the rest of the experiment, we keep the value of θ as 0.05. Note that in the permanence-based post-processing step, we only consider those vertices having at least one external connection. The rationale behind this assumption is that vertices in the core of each community are

J. Stat. Mech. (2015) P05017

Algorithm 1 PVOC: Permanence based vertex-replication algorithm for overlapping community detection

Input: A graph $G = (V, E)$; A_d= disjoint community detection algorithm; θ =threshold
Output: Detected overlapping communities

procedure VERTEX_REPLICATION(G, NC, θ)
  V_e = set of vertices having at least one external neighbor
  for all do $v \in V_e$
    C_v = current community of v
    Measure current permanence of v, $O_p(v)$
    Measure current sum of permanences of all neighbor's of v, $O_n(v)$
    $Sum\_O_p = O_p(v) + O_n(v)$
    for all do $n \in N$ ▷ N is the set of external neighbors of v
      C_n =current community of n
      Remove v from C_v and assign it in C_n
      Measure new permanence of v in community C_n, $N_p(v)$
      Measure new sum of permanences of all neighbor's of v, $N_n(v)$
      $Sum\_N_p = N_p(v) + N_n(v)$
      if $(|Sum\_N_p - Sum\_O_p| <= \theta)$ **then**
        Assign a replica of v in C_n along with its original presence in C_v
      else
        Remove vertex v from C_n and place it back in C_v
  return The updated overlapping community structure
procedure PVOC
  Run A_d on G to obtain disjoint community structure NC
  Call VERTEX_REPLICATION(G,NC,θ)

often considered to be correctly placed by the disjoint community detection algorithm, whereas vertices which are placed in the peripheral region of the community and are loosely connected to the core of the community have high chance to be part of multiple communities. Figure 5 shows an empirical observation where we plot the relation between the number of external connections of a vertex in the detected disjoint community to the number of overlapping memberships of the vertex in ground-truth community. We observe that the correlation is increasing in nature, which indeed strengthens our hypothesis.

The time complexity of measuring the permanence of a vertex takes $O(d^2)$, where d is the average degree of vertices in the network. In real-world networks, the value of d is much lower than $\log(n)$, where n is the number of nodes in the network. Therefore, the PVOC algorithm mostly depends on the underlying disjoint community detection algorithm.

5. Experiments

We combine PVOC with two popular disjoint community detection algorithms, namely Louvain[3] [10] and Infomap[4] [11, 46]. These are chosen because they are reasonably

[3] https://sites.google.com/site/findcommunities/.
[4] www.tp.umu.se/~rosvall/code.html.

J. Stat. Mech. (2015) P05017

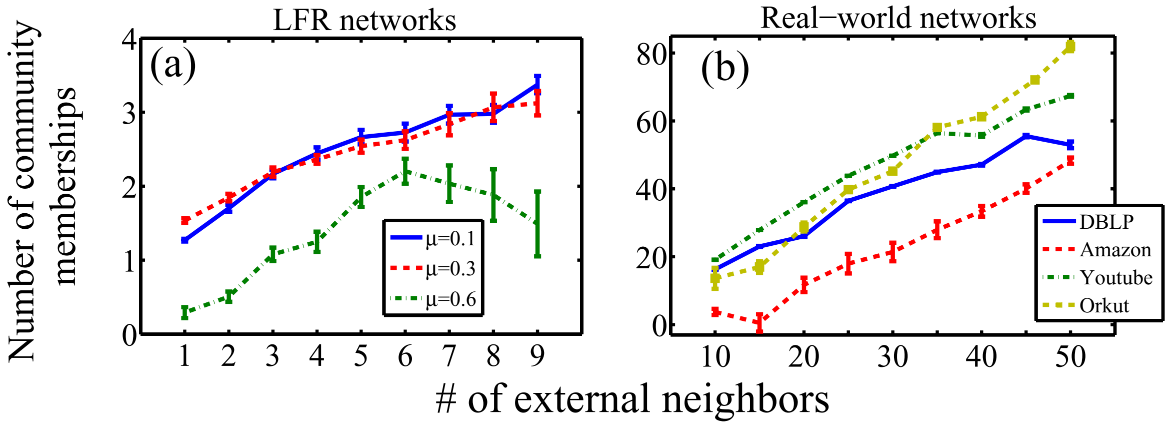


Figure 5. The relation between the average number of external connections of vertices (with the standard deviation) obtained from the output of the disjoint community detection algorithm (here we use Louvian) and the number of communities a vertex is a part of.

accurate algorithms with the potential to handle large networks, and implementation of them, by their authors, are publicly available.

5.1. Baseline algorithms

We compare the performance of PVOC with the following state-of-the-art overlapping community detection algorithms, whose codes are also available:

- Order statistics local optimization method (OSLOM): It is based on the local optimization of a fitness function expressing the statistical significance of clusters with respect to random fluctuations, which is estimated with tools of Extreme and Order Statistics [5]. The code is available at http://www.oslom.org.

- Community overlap propagation algorithm (COPRA): This algorithm is based on the label propagation technique of Raghavan *et al* [4], but is able to detect communities that overlap. Like the original algorithm, vertices have labels that propagate between neighboring vertices so that members of a community reach a consensus on their community membership [36]. The code is available at www.cs.bris.ac.uk/∼steve/networks/software/copra.html.

- Speaker listener propagation algorithm (SLPA): The algorithm is an extension of the label propagation algorithm (LPA) [4]. In SLPA, each node can be a listener or a speaker. The roles are switched depending on whether a node serves as an information provider or information consumer. Typically, a node can hold as many labels as it likes, depending on what it has experienced in the stochastic processes driven by the underlying network structure. A node accumulates knowledge of repeatedly observed labels instead of erasing all but one of them. Moreover, the more a node observes a label, the more likely it will spread this label to other nodes [37]. The code is available at https://sites.google.com/site/communitydetectionslpa.

- Agglomerative hierarchical clustering based on maximal clique (EAGLE): It uses the agglomerative framework to produce a dendrogram. First, all maximal cliques are found and made to be the initial communities. Then, the pair of communities with maximum similarity is merged. The optimal cut on the dendrogram is determined by the extended modularity with a weight based on the number of overlapping memberships [6]. The code is available at http://code.google.com/p/eaglepp/.

- Cluster-overlap Newman Griven algorithm (CONGA): The idea of this algorithm is similar to our idea of finding overlapping communities from disjoint community structure. CONGA is based on Griven Newman's 'GN' algorithm [1] but extended to detect overlapping communities. CONGA adds to the GN algorithm the ability to split vertices between communities, based on the new concept of *split betweenness*. At first, edge betweenness of edges and split betweenness of vertices are calculated. Then an edge with maximum edge betweenness is removed or a vertex with maximum split betweenness is split. After this step, edge betweenness and split betweenness are recalculated. The above steps are repeated until no edges remain [7]. However, the calculation of edge betweenness and split betweenness is expensive on large networks. The code is available at www.cs.bris.ac.uk/~steve/networks/congapaper/.

- Cluster affiliation model for big networks (BIGCLAM): In this algorithm, communities arise due to shared community affiliations of nodes. Here the affiliation strength is explicitly modeled for each node to each community. Then each node-community pair is assigned a nonnegative latent factor which represents the degree of membership of a node to the community. The probability of an edge between a pair of nodes is then modeled in the network as a function of the shared community affiliations [35]. The code is available at http://snap.stanford.edu.

Note that each algorithm is simply used with its default parameters.

5.2. Validation metrics

A stronger test of the correctness of the community detection algorithm, however, is by comparing the obtained community with a given ground–truth structure. For evaluation, we use three metrics that quantify the level of correspondence between the detected and the ground–truth communities [35].

- Overlapping Normalized Mutual Information[5] [48]

- Omega Index [24]

- Average F score [49]

Note that all the metrics are bounded between 0 (no matching) and 1 (perfect matching).

[5] https://github.com/aaronmcdaid/Overlapping-NMI.

J. Stat. Mech. (2015) P05017

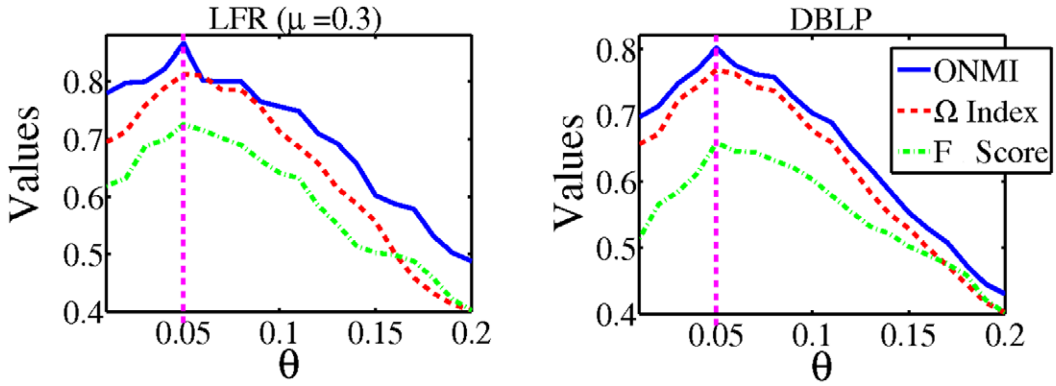


Figure 6. Accuracy of PVOC (in terms of three validation metrics) with the increase of θ for LFR ($\mu = 0.3$) and one real-world network (DBLP). Maximum accuracy is obtained at $\theta = 0.05$, which we use in rest of the experiment. Each point in the plot is an average of the accuracies obtained from Louvain and Infomap.

5.3. Experimental results

In this experiment, we use PVOC combined with Louvain and Informap separately, and compare the results with six baseline algorithms. First, we check the dependency of PVOC with the value of θ. Figure 6 shows that at $\theta = 0.05$, PVOC achieves maximum accuracy for LFR and one representative real-world network; however the result is almost same of other networks. Therefore, we use $\theta = 0.05$ in the rest of the experiments. One can tune θ appropriately to control the extent of overlapping membership of vertices in the network.

In figure 7, we compare the outputs obtained from different competing algorithms with the ground-truth communities for LFR networks with different parameter settings. Figure 7 (top panel) shows the results for different values of μ ranging from 0.1 to 0.5. As μ increases, the community structure becomes less evident and it becomes difficult for all the algorithms to discover the actual community structure. OSLOM performs worst compared to the other algorithms. However, for all the cases, PVOC+LVN is least affected and outperforms other algorithms. This is followed by PVOC+INFO, CONGO and SLPA.

We then vary the average number of community memberships per vertex, O_m from 2 to 8 keeping the other parameters same, and plot the performance of different algorithms in figure 7 (middle panel). The effect is reasonably less on the accuracy of the competing algorithms. Here we observe that the pattern is almost similar for PVOC+LVN and PVOC+INFO, and are much superior than others.

Finally, in figure 7 (lower panel) we plot the accuracy of the algorithms with the increasing value of O_n, percentage of overlapping vertices. Surprisingly, OSLOM shows an unexpected behavior with the increasing accuracy after a certain value of O_n. However, on an average the change in accuracy is almost consistent for all the algorithms in all possibilities of O_n.

To understand the utility of including PVOC step with the disjoint community finding algorithms in more details, we further measure the performance of Louvain and Infomap in isolation without PVOC step. We observe that excluding PVOC step significantly deteriorates the performance of Louvain algorithm: for LFR network ($N = 10\,000$,

J. Stat. Mech. (2015) P05017

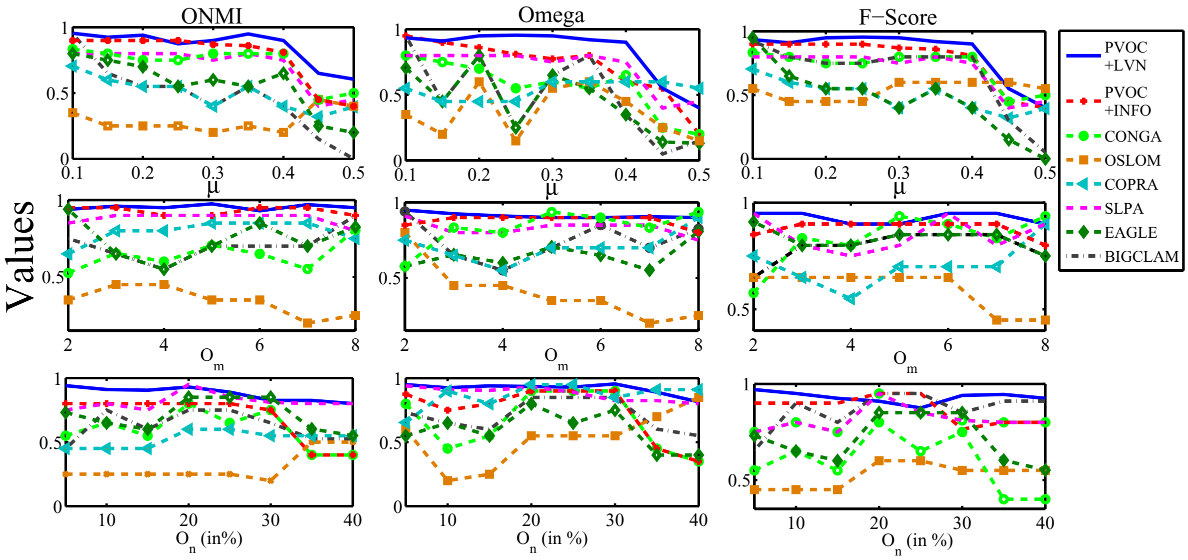


Figure 7. Accuracy of all the competing algorithms for LFR by varying μ (top panel, where $N = 10\,000$, $O_m = 4$, $O_n = 5\%$), O_m (middle point, where $N = 10\,000$, $\mu = 0.1$, $O_n = 5\%$) and O_n (bottom panel, where $N = 10\,000$, $\mu = 0.1$, $O_m = 4$)). Note that the value of O_n is expressed in % of n (LVN: Louvain, INFO: Infomap).

$O_m = 4$, $O_n = 5\%$, $\mu = 0.2$) ONMI (0.569), Omega Index (0.512), F-Score (0.523); for DBLP network ONMI (0.495), Omega Index (0.521), F-Score (0.487); for Amazon network ONMI (0.458), Omega Index (0.498), F-Score (0.447); for Youtube network ONMI (0.512), Omega Index (0.522), F-Score (0.564); and for Orkut network ONMI (0.526), Omega Index (0.556), F-Score (0.544). Similar trend is observed for Infomap algorithm. This observation therefore strengthens the need of PVOC as a post-processing step with the disjoint community detection algorithms.

Now, we run the competing algorithms on the real-world networks. As noted in [35], most of the baseline community detection algorithms do not scale for networks of large size. Therefore, we use the following technique proposed by Yan and Leskovec [35] to obtain several small subnetworks with overlapping community structure from the large real networks. We pick a random node u in the given graph G that belongs to at least two communities. We then take the subnetwork to be the induced subgraph of G consisting of all the nodes that share at least one ground-truth community membership with u. In our experiments, we created 500 different subnetworks for each of the six real-world datasets and the results are averaged over these 500 samples. For each validation metric (ONMI, Ω Index, F-Score), we separately scale the scores of the methods so that the best performing community detection method has the score of 1. Finally, we compute the composite performance by summing up three normalized scores. If a method outperforms all the other methods in all the scores, then its composite performance is 3.

Figure 8 displays the composite performance of the methods for different networks. On an average, the composite performance of PVOC+INFO (2.88) and PVOC+LVN (2.74) significantly outperform other competing algorithms: 6.27% higher than that of BIGCLAM (2.71), 18.03% higher than that of SLPA (2.44), 101.3% higher than that

J. Stat. Mech. (2015) P05017

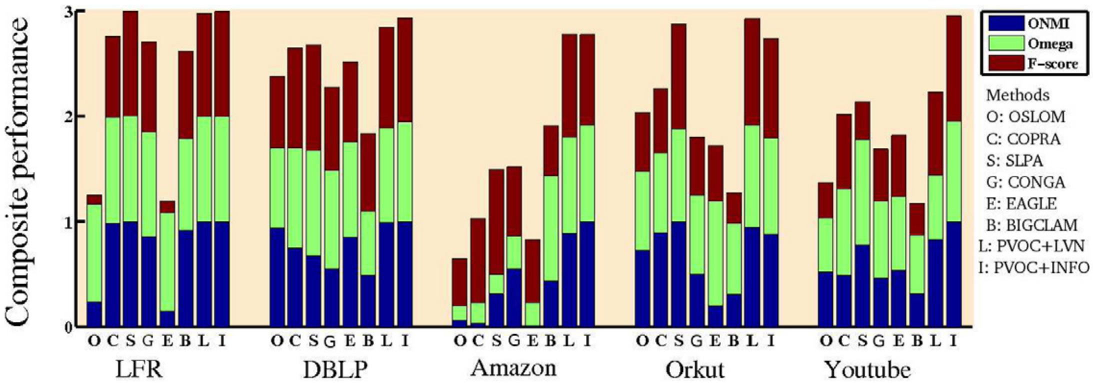


Figure 8. Performance of various competing algorithms to detect the ground-truth communities. For each evaluation metric separately we scale the score of the methods so that the best performing community detection algorithm achieves the score of 1. Thus if an algorithm outperforms all the methods in all the scores, then its composite score would become 3.

Table 3. The performance of BIGCLAM and PVOC on large real-world networks.

	BIGCLAM			PVOC+LVN			PVOC+INFO		
Networks	ONMI	Omega	F Score	ONMI	Omega	F Score	ONMI	Omega	F Score
DBLP	0.61	0.59	0.54	**0.65**	0.61	**0.60**	**0.65**	**0.62**	0.59
Amazon	**0.73**	0.69	0.74	0.72	0.71	0.75	**0.73**	**0.74**	**0.76**
Orkut	0.65	0.68	0.64	0.72	0.70	0.76	**0.73**	**0.72**	**0.77**
Youtube	0.68	0.76	**0.78**	**0.77**	**0.78**	0.72	0.71	0.68	**0.78**

of OSLOM (1.43), 36.4% higher than that of COPRA (2.11), 48.4% higher than that of CONGA (1.94), and 77.8% higher than that of EAGLE (1.62). The absolute average ONMI of PVOC+INFO (PVOC+LVN) for one LFR and six real networks taken together is 0.85 (0.83), which is 4.93% (2.46%) and 26.8% (20.8%) higher than the two most competing algorithms, i.e. BIGCLAM (0.81), and SLPA (0.67) respectively. In terms of absolute values of scores, PVOC+INFO (PVOC+LVN) achieves the average F-Score of 0.84 (0.79) and average Ω Index of 0.83 (0.82). Overall, PVOC combined with Louvain and Infomap gives the best results, followed by BIGCLAM, SLPA, COPRA, CONGO, EAGLE and OSLOM.

As most of the baseline algorithms except BIGCLAM do not scale for large real networks [35], we separately compare PVOC with BIGCLAM (which is scalable and also the most competing algorithm) on actual large real datasets. Table 3 shows performance of PVOC and BIGCLAM for different real networks. On average, PVOC+INFO (PVOC+LVN) achieves 4.28% (5.63%) higher ONMI, 1.48% (2.85%) higher Ω Index, and 6.94% (5.63%) higher F-Score. Overall, PVOC outperforms BIGCLAM in every measure and for every network. The absolute values of the scores of PVOC+INFO and PVOC+LVN averaged over all the networks are 0.70 and 0.71 (ONMI), 0.69 and 0.70 (Ω Index), and 0.72 and 0.71 (F-Score) respectively.

Many optimization algorithms have the tendency to underestimate smaller size communities [50] and sometimes tend to produce very large size communities. In our test

J. Stat. Mech. (2015) P05017

Table 4. Size of the largest and smallest communities present in the ground-truth and that obtained from BIGCLAM and PVOC based algorithms (the Jaccard similarities between the results obtained from the algorithms with the ground-truth structure are reported within parenthesis) for both LFR and real-world networks.

	Ground-truth		BIGCLAM		PVOC+LVN		PVOC+INFO	
Networks	Max Size	Min size	Max Size	Min size	Max Size	Min size	Max Size	Min size
DBLP	3458	124	9876 (0.56)	877 (0.48)	4098 (0.71)	243 (0.82)	4143 (0.76)	204 (0.81)
Amazon	5987	245	10 109 (0.45)	765 (0.57)	6876 (0.69)	398 (0.75)	6367 (0.72)	323 (0.83)
Orkut	10 687	1876	13 768 (0.72)	2985 (0.69)	11 976 (0.74)	1908 (0.79)	11 345 (0.75)	1976 (0.79)
Youtube	8987	765	9976 (0.65)	1098 (0.62)	8876 (0.76)	987 (0.71)	9018 (0.74)	865 (0.82)

suite, we observe the similar tendency in BIGCLAM whereas the communities obtained by PVOC based algorithms are comparable in size with respect to the ground-truth. Earlier in table 2, we have mentioned the number of communities detected by PVOC based algorithms (the number of communities does not change due to the inclusion of PVOC step with Louvain and Infomap). In table 4, we show for both LFR and real-world networks that the size of the largest and smallest communities detected by BIGCLAM is much larger than that present in the ground-truth structure. We also measure the similarity (using Jaccard coefficient) between the largest and smallest-size communities detected by BIGCLAM and PVOC based algorithms with the communities in ground-truth structure and notice that PVOC based algorithms are able to detect both largest and smallest-size communities which are most similar to the ground-truth structure. Therefore, we hypothesize that our algorithm has the potentiality to produce meaningful communities which have high resemblance with the ground-truth structure.

6. Conclusions

In this paper, we presented a study to show that there is perhaps less need of developing yet another algorithm for finding overlapping communities from the network. We demonstrated how the output of an efficient disjoint community detection algorithm can be leveraged to discover the overlapping community structure. For that, we proposed a novel, two-phase framework, called PVOC that can be combined with any efficient disjoint community detection algorithm. PVOC uses a new metric, called permanence in the post-processing step on each vertex and detects the overlapping vertices from the non-overlapping structure. We combined PVOC with two efficient and scalable algorithms, Louvain and Informap. Experimental results showed that our approach is viable in producing meaningful overlapping communities quite efficiently even from the large real world networks in terms of high resemblance with the ground-truth community structure. PVOC is controlled by only one parameter θ, which can be efficiently tuned to increase the extent of overlapping memberships per vertex in a network.

However, a major drawback of PVOC is that it produces exactly the same number of overlapping communities that the disjoint community detection algorithm produces.

However, it might be possible that due to the overlapping nature of a community, new community might emerge from the disjoint community structure. As an immediate step, we would like to include a new module in the post-processing step that would consider the emergence of new communities. Moreover, we would try to evaluate PVOC in conjunction with even more disjoint community detection algorithms. To conclude, we would like to emphasize on the fact that considering such a massive literature particularly on community detection, it is perhaps the good time to put an end to such consistent effort of proposing yet another algorithm, and to revisit some of the existing algorithms that are efficient enough to fulfill both the purpose of discovering disjoint and overlapping communities from the network.

Acknowledgments

The author would like to thank Niloy Ganguly and Animesh Mukherjee for their contributions.

References

[1] Newman M E J and Girvan M 2004 *Phys. Rev.* E **69** 026113
[2] Fortunato S 2010 *Phys. Rep.* **486** 75–174
[3] Papadopoulos S, Kompatsiaris Y, Vakali A and Spyridonos P 2012 *Data Min. Knowl. Discovery* **24** 515–54
[4] Raghavan U N, Albert R and Kumara S 2007 *Phys. Rev.* E **76** 036106+
[5] Lancichinetti A, Radicchi F, Ramasco J J and Fortunato S 2011 *PLoS ONE* **6** e18961
[6] Shen H and Hu M B 2009 *Physica* A **388** 1706–12
[7] Gregory S 2007 An algorithm to find overlapping community structure in networks *Proc. of the 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases* (Berlin: Springer) pp 91–102
[8] Xie J, Kelley S and Szymanski B K 2013 *ACM Comput. Surv.* **45** 43:1–43:35
[9] Chakraborty T, Srinivasan S, Ganguly N, Mukherjee A and Bhowmick S 2014 On the permanence of vertices in network communities *The 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York, USA, 24–27 August 2014)* pp 1396–1405
[10] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 *J. Stat. Mech.* P10008
[11] Rosvall M and Bergstrom C 2007 *Proc. Natl Acad. Sci.* **104** 7327
[12] Palla G, Dernyi I, Farkas I and Vicsek T 2005 *Nature* **435** 814–8
[13] Fortunato S and Lancichinetti A 2009 Community detection algorithms: A comparative analysis: Invited presentation, extended abstract *Proc. Fourth Int. ICST Conf. on Performance Evaluation Methodologies and Tools (Brussels, Belgium)* (Belgium: Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) pp 27:1–27:2 http://dl.acm.org/citation.cfm?id=1698858&dl=ACM&coll=DL
[14] Ahn Y Y, Bagrow J P and Lehmann S 2010 *Nature* **466** 761–4
[15] Evans T S and Lambiotte R *Eur. Phys. J.* B **77** 265–72
[16] Evans T S and Lambiotte R 2009 *Phys. Rev.* E **80** 016105
[17] Chen Y, Wang X L, Yuan B and Tang B Z 2014 *J. Stat. Mech.* P03021
[18] Baumes J, Goldberg M K, Krishnamoorthy M S, Magdon-Ismail M and Preston N 2005 Finding communities by clustering a graph into overlapping subgraphs *IADIS AC* pp 97–104
[19] Lancichinetti A, Fortunato S and Kertész J 2009 *New J. Phys.* **11** 033015
[20] Havemann F 0003 M H, Struck A and Glser J 2011 *J. Stat. Mech.* P01023
[21] Chen D, Shang M, Lv Z and Fu Y 2010 *Phys. A: Stat. Mech. Appl.* **389** 4177–87
[22] Lee C, Reid F, McDaid A and Hurley N 2010 Detecting highly-overlapping community structure by greedy clique expansion *Workshop: ACM KDD-SNA* pp 33–42 (arXiv:1002.1827)
[23] Du N, Wang B and WU B 2008 Overlapping community structure detection in networks *17th ACM Conf. on Information and Knowledge Management* pp 1371–2

[24] Gregory S 2011 *J. Stat. Mech.* P02017
[25] Nepusz T, Petroczi A, Negyessy L and Bazso F 2008 *Phys. Rev.* E **77** 016107
[26] Zhang S, Wang R S and Zhang X S 2007 *Phys. A: Stat. Mech. Appl.* **374** 483–90
[27] Newman M E J and Leicht E A 2007 *Proc. Natl Acad. Sci. USA* **104** 9564–9
[28] Ren W, Yan G, Liao X and Xiao L 2009 *Phys. Rev.* E **79** 036111
[29] Nowicki K and Snijders T A B 2001 *J. Am. Stat. Assoc.* **96** 1077–87
[30] Zarei M, Izadi D and Samani K A 2009 *J. Stat. Mech.* P11013
[31] McDaid A and Hurley N 2010 Detecting highly overlapping communities with model-based overlapping seed expansion *Proc. 2010 Int. Conf. on Advances in Social Networks Analysis and Mining (Washington, DC, USA)* pp 112–119
[32] Zhang S, Wang R S and Zhang X S 2007 *Phys. Rev.* E **76** 046103
[33] Zhao K, Zhang S W and Pan Q 2010 Fuzzy analysis for overlapping community structure of complex network *Chinese Control and Decision Conf.* pp 3976–81
[34] Ding F, Luo Z, Shi J and Fang X 2010 Overlapping community detection by kernel-based fuzzy affinity propagation *Int. Workshop on Indoor Spatial Awareness* pp 1–4
[35] Yang J and Leskovec J 2013 Overlapping community detection at scale: a nonnegative matrix factorization approach *WSDM* (New York: ACM) pp 587–96
[36] Gregory S 2010 *New J. Phys.* **12** 103018
[37] Xie J and Szymanski B K 2012 Towards linear time overlapping community detection in social networks *PAKDD* pp 25–36 (arXiv:1202.2465)
[38] Xie J and Szymanski B K 2011 Community detection using a neighbourhood strength driven label propogation algorithm *IEEE NSW 2011* pp 188–95
[39] Chen W, Liu Z, Sun X and Wang Y 2010 *Data Min. Knowl. Discovery* **21** 224–40
[40] Girvan M and Newman M E J 2002 *Proc. Natl Acad. Sci. USA* **99** 7821–6
[41] Zhang Y, Wang J, Wang Y and Zhou L 2009 Parallel community detection on large networks with propinquity dynamics *15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 997–1006
[42] István A, Palotai R, Szalay M S and Csermely P K 2010 *PLoS ONE* **5** e12528
[43] Gopalan P K and Blei D M 2013 *Proc. Natl Acad. Sci. USA* **110** 14534–9
[44] Lancichinetti A, Fortunato S and Radicchi F 2008 *Phys. Rev.* E **78** 046110
[45] Yang J and Leskovec J 2012 Defining and evaluating network communities based on ground-truth *Proc. ACM SIGKDD Workshop on Mining Data Semantics* (New York: ACM) pp 3:1–3:8
[46] Rosvall M and Bergstrom C T 2008 *Proc. Natl Acad. Sci. USA* **105** 1118–23
[47] Danon L, Diaz-Guilera A, Duch J and Arenas A 2005 *J. Stat. Mech.* P09008
[48] McDaid A F, Greene D and Hurley N J 2011 arXiv:1110.2515
[49] Manning C D, Raghavan P and Schütze H 2008 *Introduction to Information Retrieval* (Cambridge: Cambridge University Press)
[50] Fortunato S and Barthelemy M 2007 *Proc. Natl Acad. Sci. USA* **104** 36–41

J. Stat. Mech. (2015) P05017