Overlapping Community Detection in Networks: The State-of-the-Art and Comparative Study

Article *in* ACM Computing Surveys · October 2011

DOI: 10.1145/2501654.2501657 · Source: arXiv

Overlapping Community Detection in Networks: the State of the Art and Comparative Study

JIERUI XIE
Rensselaer Polytechnic Institute
and
STEPHEN KELLEY
Oak Ridge National Laboratory
and
BOLESLAW K. SZYMANSKI
Rensselaer Polytechnic Institute

This paper reviews the state of the art in *overlapping* community detection algorithms, quality measures, and benchmarks. A thorough comparison of different algorithms is provided. In addition to *community level* evaluation, we propose a novel framework for evaluating algorithms' ability to detect *overlapping nodes*, which helps to assess *overdetection* and *underdetection*. We conclude that SLPA, OSLOM, Game, and COPRA offer the best performance. A common feature observed by various algorithms in real-world networks is the relatively small fraction of overlapping nodes (typically less than 30%), each of which belongs to only 2 or 3 communities.

Categories and Subject Descriptors: A.1 [**General Literature**]: INTRODUCTORY AND SURVEY; I.5.3 [**Clustering**]: Clustering—*Algorithms*; H.3.3 [**Clustering**]: Information Search and Retrieval—*Clustering*; E.1 [**Data**]: DATA STRUCTURES—*Graphs and networks*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Algorithms, overlapping community detection, social networks

1. INTRODUCTION

Community or modular structure is considered to be a significant property of real-world social networks as it often accounts for the functionality of the system. Despite the ambiguity in the definition of *community*, numerous techniques have been developed for both efficient and effective community detection. Random walks, spectral clustering, modularity maximization, differential equations, and statistical mechanics have all been used previously. Much of the focus within community detection has been on identifying *disjoint* communities. This type of detection assumes that the network can be partitioned into dense regions in which nodes have more connections to each other than to the rest of the network. Recent reviews on disjoint community detection are presented in [Fortunato 2010; Danon et al. 2005; Lancichinetti and Fortunato 2009b; Leskovec et al. 2010].

However, it is well understood that people in a social network are naturally characterized by *multiple* community *memberships*. For example, a person usually has connections to several social groups like family, friends, and colleagues; a researcher may be active in several areas. Further, in online social networks, the number of communities an individual can belong to is essentially unlimited because a person can simultaneously associate with as many groups as he wishes. This also happens in other complex networks such as biological networks, where a node might have multiple functions. In [Kelley et al. 2011], the authors showed that the *overlap* is indeed a significant feature of many real social networks.

For this reason, there is growing interest in overlapping community detection algorithms that identify a set of clusters that are not necessarily disjoint. There could be nodes that belong to more than one cluster. In this paper, we offer a review on the state of the art.

2. PRELIMINARIES

In this section, we present basic definitions that will be used throughout the paper. Given a network or graph $G = \{E, V\}$, V is a set of n nodes and E is a set of m edges. For dense graphs $m = O(n^2)$, but for sparse networks $m = O(n)$. The network structure is determined by the $n \times n$ adjacency matrix A for unweighted networks and weight matrix W for weighted networks. Each element A_{ij} of A is equal to 1 if there is an edge connecting nodes i and j; and it is 0 otherwise. Each element w_{ij} of W takes a nonnegative real value representing strength of connection between nodes i and j.

In the case of overlapping community detection, the set of clusters found is called a *cover* $C = \{c_1, c_2, \cdots, c_k\}$ [Lancichinetti et al. 2009], in which a node may belong to more than one cluster. Each node i associates with a community according to a *belonging factor* (i.e., soft assignment or membership) $[a_{i1}, a_{i2}, \cdots, a_{ik}]$ [Nepusz et al. 2008], in which a_{ic} is a measure of the strength of association between node i and cluster c. Without loss of generality, the following constraints are assumed to be satisfied

$$0 \leq a_{ic} \leq 1 \;\; \forall i \in V, \forall c \in C \tag{1}$$

and

$$\sum_{c=1}^{|C|} a_{ic} = 1,$$

where $|C|$ is the number of clusters. However, the belonging factor is often solely a set of artificial weights. It may not have a clear or unambiguous physical meaning [Shen et al. 2009].

In general, algorithms produce results that are composed of one of two types of assignments, *crisp* (non-fuzzy) assignment or *fuzzy* assignment [Gregory 2011]. With crisp assignment, each node belongs to one or more communities with *equal* strength. The relationship between a node and a cluster is *binary.* That is, a node i either belongs to cluster c or does not. With fuzzy assignment, each node is associated with communities in proportion to a belonging factor. With a threshold, a fuzzy assignment can be easily converted to a crisp assignment. Typically, a detection algorithm outputs crisp community assignments.

3. ALGORITHMS

In this section, algorithms for overlapping community detection are reviewed and categorized into five classes which reflect how communities are identified.

3.1 Clique Percolation

The clique percolation algorithm (CPM) is based on the assumption that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. It begins by identifying all cliques of size k in a network. Once these have been identified, a new graph is constructed such that each vertex represents one of these k-cliques. Two nodes are connected if the k-cliques which represent them share k-1 members. Connected components in the new graph identify which cliques compose the communities. Since a vertex can be in multiple k-cliques simultaneously, overlap between communities is possible. CPM is suitable for networks with dense connected parts. Empirically, $k = 3$ or 4 has been shown to give the best results. CFinder[1] is the implementation of CPM, whose time complexity is polynomial in many applications [Palla et al. 2005]. However, it also fails to terminate in many large social networks.

CPMw [Farkas et al. 2007] introduces a subgraph intensity threshold for weighted networks. Only k-cliques with intensity larger than a fixed threshold are included into a community. Instead of processing all values of k, SCP [Kumpula et al. 2008] finds clique communities of a given size. In the first phase, SCP detects k-cliques by checking all the (k-2)-cliques in the common neighbors of two endpoints when links are inserted to the network sequentially in the order of decreasing weights. In the second phase, the k-community is detected by finding the connected components in the (k-1)-clique projection of the bipartite representation, in which one type of node represents a k-clique and the other denotes a (k-1)-clique. Since each k-clique is processed exactly twice, the running time grows linearly as a function of the number of cliques. SCP allows multiple weight thresholds in a single run and is faster than CPM.

Despite conceptual simplicity, one may argue that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structure in a network.

3.2 Line Graph and Link Partitioning

The idea of partitioning links instead of nodes to discover community structure has also been explored. A node in the original graph is called overlapping if links connected to it are put in more than one cluster.

In [Ahn et al. 2010][2], links are partitioned via hierarchical clustering of edge similarity. Given a pair of links e_{ik} and e_{jk} incident on a node k, a similarity can be computed via the Jaccard Index defined as

$$S(e_{ik}, e_{jk}) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|},$$

where N_i is the neighborhood of node i including i. Single-linkage hierarchical clustering is then used to build a link dendrogram. Cutting this dendrogram at some threshold yields link communities. The time complexity is $O(nk_{max}^2)$, where k_{max} is the maximum node degree in the network.

[1] http://www.cfinder.org/

[2] https://github.com/bagrow/linkcomm

Evans [Evans and Lambiotte 2009; 2010] projected the network into a weighted *line graph*, whose nodes are the links of the original graph. Then disjoint community detection algorithms can be applied. The node partition of a line graph leads to an edge partition of the original graph. CDAEO [Wu et al. 2010] provides a post-processing procedure to determine the extent of overlapping. Once the preliminary partitioning on the line graph is done, for a node i with $|E_{icmin}|/|E_{icmax}|$ below some predefined threshold, where $E_{icmin(cmax)}$ is the set of edges in the community with which i has the minimum (maximum) number of connections, links in E_{icmin} of the line graph are removed. This essentially reduces node i to a single membership.

Kim [Kim and Jeong 2011] extended the map equation method (also known as Infomap [M. Rosvall 2008]) to the line graph, which encodes the path of the random walk on the line network under the Minimum Description Length (MDL) principle.

Line graph has been extended to clique graph [Evans 2010], wherein cliques of a given order are represented as nodes in a weighted graph. The membership strength of a node i to community c is given by the fraction of cliques containing i which are assigned to c.

Although the link partitioning for overlapping detection seems conceptually natural, there is no guarantee that it provides higher quality detection than node based detection does [Fortunato 2010] because these algorithms also rely on an ambiguous definition of community.

3.3 Local Expansion and Optimization

Algorithms utilizing local expansion and optimization are based on growing a *natural* community [Lancichinetti et al. 2009] or a partial community. Most of them rely on a local benefit function that characterizes the quality of a densely connected group of nodes.

Baumes [Baumes et al. 2005; Baumes et al. 2005] proposed a two-step process. First, the algorithm RankRemoval is used to rank nodes according to some criterion. Then the process iteratively removes highly ranked nodes until small, disjoint cluster cores are formed. These cores serve as seed communities for the second step of the process, *Iterative Scan* (IS), that expands the cores by adding or removing nodes until a local density function cannot be improved. The proposed density function can be formally given as

$$f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c},$$

where w_{in}^c and w_{out}^c are the total internal and external weight of the community c. The worst-case running time is $O(n^2)$. The quality of discovered communities depends on the quality of seeds. Since the algorithm allows vertices to be removed during the expansion, IS has been shown to produce disconnected components in some cases. For this reason, a modified version called CIS is introduced in [Kelley 2009], wherein the connectedness is checked after each iteration. In the case that the community is broken into more than one part, only the one with the largest density is kept. CIS also develops a new fitness function

$$f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c} + \lambda e_p$$

incorporating the edge probability e_p. The parameter λ controls how the algorithm behaves in sparse areas of the network. The addition of a node needs to strike a balance between the change in the internal degree density and the change in edge density.

LFM [Lancichinetti et al. 2009] expands a community from a random seed node to form a natural community until the fitness function

$$f(c) = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\alpha} \tag{2}$$

is locally maximal, where k_{in}^c and k_{out}^c are the total internal and external degree of the community c, and α is the resolution parameter controlling the size of the communities. After finding one community, LFM randomly selects another node not yet assigned to any community to grow a new community. LFM depends significantly on the resolution parameter α. The computational complexity for a fixed α-value is roughly $O(n_c s^2)$, where n_c is the number of communities and s is the average size of communities. The worst-case complexity is $O(n^2)$.

MONC [Havemann et al. 2011] uses the modified fitness function of LFM

$$f(c) = \frac{k_{in}^c + 1}{(k_{in}^c + k_{out}^c)^\alpha},$$

which allows a single node to be considered a community by itself. This avoids violation of the principle of *locality*. The proposed fitness function enables MONC to find the range of αs (resolution parameter as in LFM) for which a set of nodes is locally optimal. Rather than numerical exploration of these α values, MONC calculates the next lowest value of α which results in further expansion and continues to expand the community. In the case that the natural community of a node i is a subset of another node, the analysis of i stops. In this way, MONC merges communities during processing and, as a result, uncovers the network faster than LFM.

OSLOM[3] [Lancichinetti et al. 2011] tests the statistical significance of a cluster [Bianconi et al. 2008] with respect to a global null model (i.e., the random graph generated by the configuration model [Molloy and Reed 1995]) during community expansion. To grow the current community, the r value is computed for each neighbor, which is the cumulative probability of having the number of internal connections equal or larger than the number of connections from a neighbor into this community in the null model. If the cumulative distribution of the *smallest* r value is smaller than a given tolerance, it is considered to be significant, and the corresponding node is added to the community. Otherwise, the second smallest r is checked and so on. OSLOM usually results in a significant number of outliers or singleton communities. The worst-case complexity in general is $O(n^2)$, while the exact complexity depends on the community structure of the underlying network being studied.

Rather than considering the original network, UEOC [Jin et al. 2011] unfolds the community of a node based on the l-step transition probability of the random walk on the corresponding *annealed* network [Newman et al. 2001], which represents an

[3] http://www.oslom.org/

ensemble of networks. After sorting nodes according to the transition probabilities in descending order, the natural community is extracted with some proper cutoff. The dominating time complexity is for calculating the transition matrix, which is $O(ln^2)$.

OCA [Padrol-Sureda et al. 2010] is based on the idea of mapping each node to a d-dimensional vector. Each *subset* of nodes S is then defined as the sum of individual vectors in this set. The fitness function is defined as the directed Laplacian on function O, where O is the squared Euclidean length of a subset vector. Like LFM, starting from some initial seeds, OCA tries to remove or add a node that results in the largest increase in the value of the fitness function. OCA requires finding the most negative eigenvalue of the adjacency matrix.

Chen [Chen et al. 2010] proposed selecting a node with maximal node strength based on two quantities $B(u, c)$ (called belonging degree) and the modified modularity Qo. $B(u, c)$ measures how tightly a node u connects to a given community c compared to the rest of the network. Given two thresholds B^U and B^L, when expanding a community c, neighboring nodes with $B(u, c) > B^U$ are included in c. For nodes with $B^L \leq B(u, c) \leq B^U$, if Qo increases after adding such a node, u is added to c. The drawbacks of this algorithm are the rather arbitrary selection of the B^U and B^L thresholds and the expensive computation of Qo whose complexity is $O(kn^2)$, where k is the number of communities.

iLCD[4] [Cazabet et al. 2010] is capable of detecting both static and temporal communities. Given a set of edges created at some time step, iLCD updates the existing communities by adding a new node if its number of second neighbors (EMSN) and number of robust second neighbors (EMRSN) are greater than expected values. New edges are also allowed to create a new community if the minimum pattern is detected. Defining the similarity between two communities as the ratio of nodes in common, a merging procedure is performed to improve the detection quality if the similarity is high. iLCD relies on two parameters for adding a node and merging two communities. The complexity of iLCD is $O(nk^2)$ in general, whose precise quantity depends on community structures and its parameters.

Seeds are very important for many local optimization algorithms. A clique has been shown to be a better alternative over an individual node as a seed, serving as the basis for a wide range of algorithms. EAGLE [Shen et al. 2009; Shen et al. 2009] 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 in [Shen et al. 2009]. Even without taking into account the time required to find all the maximal cliques, EAGLE is still computationally expensive with complexity $O(n^2 + (h + n)s)$, where s is the number of maximal cliques whose upper bound is $3^{n/3}$ (i.e., theoretically exponential) [Moon and Moser 1965], and h is the number of pairs of maximal cliques which are neighbors.

Similar to EAGLE, GCE[5] [Lee et al. 2010] identifies maximum cliques as seed communities. It expands these seeds by greedily optimizing a local fitness function.

[4] `http://cazabetremy.fr/Cazabet_remy/iLCD.html`

[5] `https://sites.google.com/site/greedycliqueexpansion/`

GCE also removes communities that are similar to previously discovered using distance between communities c_1 and c_2 defined as

$$1 - \frac{|c_1 \cap c_2|}{min(|c_1|, |c_2|)}.$$

If this distance is shorter than a parameter ϵ, the communities are similar. The time complexity for greedy expansion is $O(mh)$, where m is the number of edges, and h is the number of cliques.

In COCD [Du et al. 2008], cores are a set of independent maximal cliques induced on each vertex. Two maximal cliques are said to be dependent if their *closeness* function is positive. This function is a product of the differences between the size of internal links between two maximal cliques and the number of links connecting nodes appearing only in one of the two maximal cliques. Once the cores are identified, the remaining nodes are attached to cores with which they have maximum connections. COCD runs in $O(C_{max} \cdot Tri^2)$ in the worst case, where C_{max} is the maximum size of the detected communities, and Tri is the number of triangles, whose lower bound is $\frac{9mn-2n^3-2(n^2-3m)^{3/2}}{27}$ [Fisher 1989] or $O(n^3)$ for a dense enough graph.

3.4 Fuzzy Detection

Fuzzy community detection algorithms quantify the strength of association between all pairs of nodes and communities. In these algorithms, a soft membership vector, or belonging factor [Gregory 2010], is calculated for each node. A drawback of such algorithms is the need to determine the dimensionality k of the membership vector. This value can be either provided as a parameter to the algorithm or calculated from the data.

Nepusz [Nepusz et al. 2008] modeled the overlapping community detection as a nonlinear constrained optimization problem which can be solved by simulated annealing methods. The objective function to minimize is

$$f = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\tilde{s_{ij}} - s_{ij})^2, \tag{3}$$

where w_{ij} denotes the predefined weight, $\tilde{s_{ij}}$ is the *prior* similarity between nodes i and j, and the similarity s_{ij} is defined as

$$s_{ij} = \sum_{c} a_{ic} a_{jc}, \tag{4}$$

where the variable a_{ic} is the fuzzy membership of node i in community c, subject to the total membership degree constraint in (1) and a non-empty community constraint. To determine the number of communities k, the authors increased the value of k until the community structure does not improve as measured by a modified fuzzy modularity.

Zhang [Zhang et al. 2007] proposed an algorithm based on the spectral clustering framework [Newman 2006; White and Smyth 2005]. Given an upper bound on the number of communities k, the top $k-1$ eigenvectors are computed. The network is then mapped into a d-dimensional Euclidean space, where $d \le k-1$. Instead of using k-means, fuzzy c-means (FCM) is used to obtain a soft assignment. Both detection accuracy and computation efficiency rely on the user specified value k. With running time $O(mkh + nk^2h + k^3h) + O(nk^2)$, where m is the number of edges, n is the number of nodes, the first term is for the implicitly restarted Lanczos method, and the second term is for FCM, it is not scalable for large networks.

Due to their probabilistic nature, mixture models provide an appropriate framework for overlapping community detection [Newman and Leicht 2007]. In general, the number of mixture models is equal to the number of communities, which needs to be specified in advance. In SPAEM[6] [Ren et al. 2009], the mixture model is viewed as a generative model for the links in the network. Suppose that π_r is the probability of observing community r and community r selects node i with probability $B_{r,i}$. For each r, $B_{r,i}$ is a multinomial across elements $i = 1, 2, \cdots, n$, where n is the number of nodes. Therefore, $\sum_{i=1}^{n} B_{r,i} = 1$. The edge probability e_{ij} generated by such finite mixture model is given by

$$p(e_{ij}|\pi, B) = \sum_{r=1}^{k} \pi_r B_{r,i} B_{r,j}.$$

The total probability over all the edges present in the network is maximized by the Expectation-Maximization (EM) algorithm. As in [Kim and Jeong 2011], the optimal number of communities k is identified based on the minimum description length. There is another algorithm called FOG[7] [Davis and Carley 2008] also trying to infer groups based on link evidence.

Similar mixture models can also be constructed as a generative model for nodes [Fu and Banerjee 2008]. In SSDE[8] [Magdon-ismail and Purnell 2011], the network is first mapped into a d-dimensional space using the spectral clustering method. A Gaussian Mixture Model (GMM) is then trained via Expectation-Maximization algorithm. The number of communities is determined when the increase in log-likelihood of adding a cluster is not significantly higher than that of adding a cluster to random data which is uniform over the same space.

Stochastic block model (SBM) [Nowicki and Snijders 2001] is another type of generative model for groups in the network. Fitting an empirical network to a SBM requires inferring model parameters similar to GMM. In OSBM [Latouche et al. 2011], each node i is associated with a latent vector (i.e., community assignment) Z_i with Q independent Boolean variables $Z_{iq} \in \{0, 1\}$, where Q is the number of communities, and Z_{iq} is drawn from a multivariate Bernoulli distribution. Z is inferred by maximizing the posterior probability conditioned on the present of edges as in [Ren et al. 2009]. OSBM requires more efforts than mixture models because the factorization in the observed condition distribution for edges given Z is in general intractable. MOSES[9] [McDaid and Hurley 2010] combines OSBM with the local optimization scheme, in which the fitness function is defined based on the observed condition distribution. MOSES greedily expands a community from edges. Unlike OSBM, no connection probability parameters are required as input.

[6] http://www.code.google.com/p/spaem

[7] http://www.casos.cs.cmu.edu/projects/ora/

[8] http://www.cs.rpi.edu/~purnej/code.php

[9] http://sites.google.com/site/aaronmcdaid/moses

The worst-case time complexity is $O(en^2)$, where e is the number of edges to be expanded.

Non-negative Matrix Factorization (NMF) is a feature extraction and dimensionality reduction technique in machine learning that has been adapted to community detection. NMF approximately factorizes the feature matrix V into two matrices with the non-negativity constraint as $V \approx WH$, where V is $n \times m$, W is $n \times k$, H is $k \times m$, and k is the number of communities provided by users. W represents the data in the reduced feature space. Each element $w_{i,j}$ in the normalized W quantifies the dependence of node i with respect to community j. In [Zhang et al. 2007], V is replaced with the diffusion kernel, which is a function of the Laplacian of the network. In [Zarei et al. 2009], V is defined as the correlation matrix of the columns of the Laplacian. This results in better performance than [Zhang et al. 2007]. In [Zhao et al. 2010], redundant constraints in the approximation are removed, reducing NMF to a problem of symmetrical non-negative matrix factorization (s-NMF). Psorakis [Psorakis et al. 2011] proposed a hybrid algorithm called Bayesian NMF[10]. The matrix V, where each element v_{ij} denotes a count of the interactions that took place between two nodes i and j, is decomposed via NMF as part of the parameter inference for a generative model similar to OSBM and GMM. Traditionally, NMF is inefficient with respect to both time and memory constraints due to the matrix multiplication. In the version of [Psorakis et al. 2011], the worst-case time complexity is $O(kn^2)$, where k denotes the number of communities.

Wang et al. [Wang et al. 2009] combined disjoint detection methods with local optimization algorithms. First, a partition is obtained from any algorithm for disjoint community detection. Communities attempt to add or remove nodes. The difference (called variance) of two fitness scores on a community, either including a node i or removing node i, is computed. The normalized variances form a fuzzy membership vector of node i.

Ding [Ding et al. 2010] employed the affinity propagation clustering algorithm [Frey and Dueck 2007] for overlapping detection, in which clusters are identified by representative exemplars. First, nodes are mapped as data points in the Euclidean space via the commute time kernel (a function of the inverse Laplacian). The similarity between nodes is then measured by the cosine distance. Affinity propagation reinforces two types of messages associated with each node, the responsibility $r(i,k)$ and the availability $a(i,k)$. The probability for assigning node i into the cluster represented by exemplar node k is computed by equation $p(i,k) = e^{\hat{r}(i,k)}$, where $\hat{r}$ is the normalized responsibility as in [Geweniger et al. 2009].

3.5 Agent Based and Dynamical Algorithms

The label propagation algorithm [Raghavan et al. 2007; Xie and Szymanski 2011], in which nodes with same label form a community, has been extended to overlapping community detection by allowing a node to have multiple labels. In COPRA[11] [Gregory 2010], each node updates its belonging coefficients by averaging the coefficients from all its neighbors at each time step in a synchronous fashion. The parameter v is used to control the maximum number of communities with which a

[10] http://www.robots.ox.ac.uk/~parg/software.html

[11] http://www.cs.bris.ac.uk/~steve/networks/software/copra.html

node can associate. The time complexity is $O(vm\log(vm/n))$ per iteration.

SLPA[12] [Xie et al. 2011] is a general speaker-listener based information propagation process. It spreads labels between nodes according to pairwise interaction rules. Unlike [Raghavan et al. 2007; Gregory 2010], where a node forgets knowledge gained in the previous iterations, SLPA provides each node with a memory to store received information (i.e., labels). The membership strength is interpreted as the probability of observing a label in a node's memory. One advantage of SLPA is that it does not require any knowledge about the number of communities. The time complexity is $O(tm)$, linear in the number of edges m, where t is a predefined maximum number of iterations (e.g., $t \geq 20$).

A game-theoretic framework is proposed in [Chen et al. 2010], in which a community is associated with a Nash local equilibrium. A gain function and a loss function are associated with each agent. The game assumes that each agent is selfish and selects to join, leave and switch communities based on its own utility. An agent is allowed to joint multiple communities to handle overlapping, so long as it results in increased utility. The time complexity to find the best local operation for an agent i is $O(|L_i| \cdot |L(N_i)| \cdot k_i)$, where L_i is the communities that agent i wants to joint, $L(N_i)$ is the set of communities that i's neighbors want to joint, and k_i is the node degree. The time takes to reach a local equilibrium is bounded by $O(m^2)$, where m is the number of edges.

A process in which particles walk and compete with each other to occupy nodes is presented in [Breve et al. 2009]. Particles represent different communities. Each node has an instantaneous ownership vector (similar to belonging factor) and a long term ownership vector. At each iteration, each particle takes either a random walk or a deterministic walk to one of its neighbors with some probability. If the random walk is performed, the visited neighbor updates its instantaneous ownership vector; otherwise, the long term ownership vector is updated. At the end of the process, the long term ownership vector is normalized to produce a soft assignment. Different from SLPA and COPRA, this algorithm takes a semi-supervised approach. It requires at least one labeled node per class.

Multi-state spin models [Reichardt and Bornholdt 2004; Lu et al. 2009], in which a spin is assigned to each node, can also be applied to community detection. One of such models is q-state Potts model [Blatt et al. 1996; Reichardt and Bornholdt 2004], where q is the number of states that a spin may takes, indicating the maximum number of communities. The community detection problem is equivalent to the problem of minimizing the Hamiltonian of the model. In the ground states (i.e., local minima of the Hamiltonian), the set of nodes with the same spin state form a community. The overlap of communities is linked to the degeneracy of the minima of the Hamiltonian [Reichardt. and Bornholdt 2006]. Although a co-appearance matrix keeps track of how frequently nodes i and j have been grouped together over multiple runs, it is not clear how to aggregate this information into overlapping communities when analyzing large networks.

Synchronization of a system that consists of coupled phase oscillators is able to uncover community structures. In such a model (e.g., the Kuramoto model) the phase of each unit evolves in time according to the predefined dynamics. The set

[12] https://sites.google.com/site/communitydetectionslpa/

of nodes with the same phase or frequency can be viewed as a community [Arenas et al. 2006] while nodes that do not match any observed dynamic behaviors can be considered overlapping nodes [Li et al. 2008]. Like methods utilizing a Potts model, such algorithms are parameter dependent.

3.6 Others

CONGA[13] [Gregory 2007] extends Girvan and Newman's divisive clustering algorithm (GN) [Girvan and Newman 2002] by allowing a node to split into multiple copies. Both *splitting betweenness*, defined by the number of shortest paths on the imaginary edge, and the conventional edge betweenness are considered. CONGA inherits the high computational complexity of GN. In a more refined version, CONGO [Gregory 2008] uses local betweenness to optimize the speed. Gregory [Gregory 2009] also proposed to perform disjoint detection algorithms on the network produced by splitting the node into multiple copies using the split betweenness.

Zhang[14] [Zhang et al. 2009] proposed an iterative process that reinforces the network topology and *propinquity* interpreted as the probability of a pair of nodes belonging to the same community. The propinquity between two vertices is defined as the sum of the number of direct links, number of common neighbors and the number of links within the common neighborhood. Given the topology, propinquity is computed. Propinquity above a certain threshold is then used to redistribute links, updating the topology. If the propinquity is large, a link is added to the network; otherwise, the link is removed. The propinquity can be used to perform micro clustering on each vertex to allow overlap.

Kovács et al. [Kovács et al. 2010] proposed an approach focusing on centrality-based influence functions. Community structures are interpreted as hills of the influence landscape. For each node i, the influence over each link $f_i(j,k)$ is computed. Links within a community should have higher influence than those linking distant areas of the network. The influence on a given link $c(j,k)$ is the sum of $f_i(j,k)$ over all nodes. The function $c(j,k)$ over each link defines the community landscape, wherein the communities are determined by local maxima and their surrounding regions.

Rees [Rees and Gallagher 2010] proposed an algorithm to extract the overlapping communities from the *egonet*, which is a subgraph including a center node, its neighbors, and the links around them. When all egonets are induced, each center is removed, creating small connected components among neighbors. Then, the center node is added back to each of these components to form so-called *friendship group*. Clearly, each center node can be in multiple friendship groups. The overlapping communities are determined by merging all friendship groups.

Inspired by OPTIC [Ankerst et al. 1999], an algorithm based on techniques from visualization is proposed in [Chen et al. 2009]. Nodes are ordered according to the reachability score (RS) with respect to a starting node. The reachability is based on the probability of the existence of a link between two nodes. By scanning through the obtained *sequence* of nodes, a community containing *consecutive* nodes with RS larger than a *community threshold* is found. Clearly, this algorithm is hard to

[13] http://www.cs.bris.ac.uk/~steve/networks/software/conga.html

[14] http://dbgroup.cs.tsinghua.edu.cn/zhangyz/kdd09/

apply to large networks and requires the introduction of a community threshold.

4. EVALUATION CRITERIA

Evaluating the quality of a detected partitioning or cover is nontrivial, and extending evaluation measures from disjoint to overlapping communities is rarely straightforward.

4.1 Comparing Two Covers

Unlike disjoint community detection, where a number of measures have been proposed for comparing *identified* partitions with the *known* partitions [Danon et al. 2005; Leskovec et al. 2010], only a few measures are suitable for a set of overlapping communities. Two most widely used measures are the normalized mutual information (NMI) and Omega Index.

4.1.1 *Normalized Mutual Information.* Lancichinetti [Lancichinetti et al. 2009] has extended the notion of normalized mutual information to account for overlap between communities. For each node i in cover C', its community membership can be expressed as a binary vector of length $|C'|$ (i.e., the number of clusters in C'). $(x_i)_k = 1$ if node i belongs to the k^{th} cluster C'_k; $(x_i)_k = 0$ otherwise. The k^{th} entry of this vector can be viewed as a random variable X_k, whose probability distribution is given by $P(X_k = 1) = n_k/n$, $P(X_k = 0) = 1 - P(X_k = 1)$, where $n_k = |C'_k|$ is the number of nodes in the cluster C'_k and n is the total number of nodes. The same holds for the random variable Y_l associated with the l^{th} cluster in cover C''. The joint probability distribution $P(X_k, Y_l)$ is defined as:

$$\begin{aligned} P(X_k = 1, Y_l = 1) &= \frac{|C'_k \cap C''_l|}{n} \\ P(X_k = 1, Y_l = 0) &= \frac{|C'_k| - |C'_k \cap C''_l|}{n} \\ P(X_k = 0, Y_l = 1) &= \frac{|C''_l| - |C'_k \cap C''_l|}{n} \\ P(X_k = 0, Y_l = 0) &= \frac{n - |C'_k \cup C''_l|}{n} \end{aligned}$$

The conditional entropy of a cluster X_k given Y_l is defined as $H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l)$. The entropy of X_k with respect to the entire vector Y is based on the best matching between X_k and any component of Y given by

$$H(X_k|Y) = min_{l \in \{1,2,\cdots,|C''|\}} H(X_k|Y_l).$$

The normalized conditional entropy of a cover X with respect to Y is

$$H(X|Y) = \frac{1}{|C'|} \sum_k \frac{H(X_k|Y)}{H(X_k)}.$$

In the same way, one can define $H(Y|X)$. Finally the NMI for two covers C' and C'' is given by

$$NMI(X|Y) = 1 - [H(X|Y) + H(Y|X)]/2. \tag{5}$$

The extended NMI is between 0 and 1, with 1 corresponding to a perfect matching. Note that this modified NMI does not reduce to the standard formulation of NMI when there is no overlap.

4.1.2 *Omega Index. Omega Index* [Collins and Dent 1988] is the overlapping version of the Adjusted Rand Index (ARI) [Hubert and Arabie 1985]. It is based on *pairs* of nodes in *agreement* in two covers. Here, a pair of nodes is considered to be in *agreement* if they are clustered in *exactly* the same number of communities (possibly none). That is, the omega index considers how many pairs of nodes belong together in no clusters, how many are placed together in exactly one cluster, how many are placed in exactly two clusters, and so on.

Let K_1 and K_2 be the number of communities in covers C_1 and C_2, respectively, the omega index is defined as [Gregory 2011; Havemann et al. 2011]

$$\omega(C_1, C_2) = \frac{\omega_u(C_1, C_2) - \omega_e(C_1, C_2)}{1 - \omega_e(C_1, C_2)}. \tag{6}$$

The unadjusted omega index ω_u is defined as

$$\omega_u(C_1, C_2) = \frac{1}{M} \sum_{j=0}^{max(K_1,K_2)} |t_j(C_1) \cap t_j(C_2)|,$$

where M equal to $n(n-1)/2$ represents the number of node pairs and $t_j(C)$ is the set of pairs that appear exactly j times in a cover C. The expected omega index in the null model ω_e is given by

$$\omega_e(C_1, C_2) = \frac{1}{M^2} \sum_{j=0}^{max(K_1,K_2)} |t_j(C_1)| \cdot |t_j(C_2)|.$$

The subtraction of the expected value in (6) takes into account agreements resulting from chance alone. The omega index takes on values in the range [0,1]. The larger the value is, the better the matching is between two covers. A value of 1 indicates perfect matching. When there is no overlap, the omega index reduces to the ARI.

In addition to NMI and Omega, some other measures have been proposed, such as the generalized external indexes [Campello 2007; Campello. 2010] and the fuzzy rand index [Hüllermeier and Rifqi 2009].

4.2 Overlapping Modularity

To measure the quality of a cover produced by overlapping detection algorithms on real-world social networks where the ground truth is usually *unknown*, most measures extend the framework of modularity Q for a disjoint partition [Newman 2004], which is given as

$$Q = \frac{1}{2m} \sum_{c} \sum_{i,j \in c} \left[A_{ij} - \frac{k_i k_j}{2m} \right], \tag{7}$$

where c is a community, A_{ij} is the element of the adjacency matrix for nodes i and j, $m = \frac{1}{2}\sum_{i,j} A_{ij}$ is the total number of edges, and k_i is the degree of node i. Q

emphasizes the internal density in each community, and the probabilities of present edges are adjusted by the probability in the null model.

4.2.1 *Modularity for Crisp Assignment.* Direct extensions of Q for crisp assignment have been proposed. Shen [Shen et al. 2009] used the number of communities to which a node belongs as a weight for Q, defining the extended modularity as

$$Q_{ov}^{E} = \frac{1}{2m} \sum_{c} \sum_{i,j \in c} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \frac{1}{O_i O_j}, \tag{8}$$

where O_i is the number of communities to which node i belongs. A more complicated version of Q_{ov}^{E} is presented in [Shen et al. 2009]. This measure is in the same form as (10), but with a coefficient defined based on the maximal clique. One may argue that they are identical as in [Gregory 2011].

Chen [Chen et al. 2010] proposed the following modularity for weighted networks

$$Q_{ov}^{C} = \frac{1}{2m} \sum_{c} \sum_{i,j \in V} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \beta_{ic} \beta_{jc}, \tag{9}$$

where $\beta_{ic} = k_{ic} / \sum_{c'} k_{ic'}$ is the strength with which node i belongs to community c, and $k_{ic} = \sum_{j \in c} w_{ij}$ is the total weight of links from i into community c.

4.2.2 *Modularity for Fuzzy Assignment.* A fuzzy version of the modularity utilizing the belonging factor was previously proposed by Nepusz [Nepusz et al. 2008], by weighting Q with the *product* of a node's belonging factor

$$Q_{ov}^{Ne} = \frac{1}{2m} \sum_{c} \sum_{i,j \in c} \left[A_{ij} - \frac{k_i k_j}{2m} \right] a_{ic} a_{jc}, \tag{10}$$

where a_{ic} is the degree of membership of node i in the community c.

Zhang [Zhang et al. 2007] used the *average* of the belonging factor to define a new modularity function as

$$Q_{ov}^{Z} = \sum_{c} \left[\frac{A(V'_c, V'_c)}{A(V,V)} - \left(\frac{A(V'_c, V)}{A(V,V)} \right)^2 \right],$$

where V'_c is the set of nodes in a community c, w_{ij} is the weight of the link connecting nodes i and j, $A(V'_c, V'_c) = \sum_{i,j \in V'_c} w_{ij}(a_{ic} + a_{jc})/2$, $A(V'_c, V) = A(V'_c, V'_c) + \sum_{i \in V'_c, j \in V \setminus V'_c} w_{ij}(a_{ic} + (1 - a_{jc}))/2$, and $A(V,V) = \sum_{i,j \in V} w_{ij}$.

Another measure that is suitable for fuzzy overlapping has recently been proposed by Lazar [Lázár et al. 2010].

4.2.3 *Link based Modularity.* Nicosia [Nicosia et al. 2009] proposed an extension to the modularity measure based on the belonging coefficients of *links*. Let a link $l(i,j)$ connecting i to j for community c be $\beta_{l(i,j),c} = F(a_{ic}, a_{jc})$, the expected belonging coefficient of any possible link $l(i,j)$ from node i to a node j in community c can be defined as $\beta_{l(i,j),c}^{out} = \frac{1}{|V|} \sum_{j \in V} F(a_{ic}, a_{jc})$. Accordingly, the expected belonging coefficient of any link $l(i,j)$ pointing to node j in community c is defined as $\beta_{l(i,j),c}^{in} = \frac{1}{|V|} \sum_{i \in V} F(a_{ic}, a_{jc})$. The above belonging coefficients are used as weights for the probability of an observed link (first term in (11)) and the probability of a link starting from i to j in the null model (second term in (11)), respectively, resulting in the new modularity defined as

$$Q_{ov}^{Ni} = \frac{1}{m} \sum_{c} \sum_{i,j \in V} \left[\beta_{l(i,j),c} A_{i,j} - \beta_{l(i,j),c}^{out} \beta_{l(i,j),c}^{in} \frac{k_i^{out} k_j^{in}}{m} \right], \tag{11}$$

where $k_i^{out(in)}$ is the number of outgoing (incoming) links of i, and m is the total number of edges. Note that Q_{ov}^{Ni} depends on the link belonging coefficient $F(a_{ic}, a_{jc})$, which could be the product, average, or maximum of a_{ic} and a_{jc}.

Similar to the original definition of modularity, the value of extended modularities is between 0 and 1. A high value indicates a significant overlapping community structure relative to the null model. However, modularity (including the extended variants) should always be used with caution due to the resolution limit beyond which no community structure can be detected [Fortunato and Barthelemy 2007]. Moreover, the complexity of the modularity landscape suggests that a partition having a high value of modularity does not imply that *true* community structure has been discovered, as such high modularity partitions also exist in random networks [Guimerà et al. 2004].

Table I. Algorithms Included in the Experiments.

Algorithm	**Reference**	**Complexity**	**Imp**
CFinder	[Palla et al. 2005]	-	C++
LFM	[Lancichinetti et al. 2009]	$O(n^2)$	C++
EAGLE	[Shen et al. 2009]	$O(n^2 + (h+n)s)$	C++
CIS	[Kelley 2009]	$O(n^2)$	C++
GCE	[Lee et al. 2010]	$O(mh)$	C++
COPRA	[Gregory 2010]	$O(vm \log(vm/n))$	Java
Game	[Chen et al. 2010]	$O(m^2)$	C++
NMF	[Psorakis et al. 2011]	$O(kn^2)$	Matlab
MOSES	[McDaid and Hurley 2010]	$O(en^2)$	C++
Link	[Ahn et al. 2010]	$O(nk_{max}^2)$	C++
iLCD	[Cazabet et al. 2010]	$O(nk^2)$	Java
UEOC	[Jin et al. 2011]	$O(ln^2)$	Matlab
OSLOM	[Lancichinetti et al. 2011]	$O(n^2)$	C++
SLPA	[Xie et al. 2011]	$O(tm)$	Java

5. BENCHMARKS

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 order to accurately perform these two analyses, networks in which the ground truth is known are needed. This requirement implies that real-world networks, which are often collected from online or observed interactions, do not paint a clear enough picture due to their lack of "ground truth". In light of this requirement, we begin our discussion with synthetic networks. In the GM benchmark [Girvan and Newman 2002], equal size communities are embedded into a network for a given expected degree and a given mixing parameter μ that measures the ratio of internal connections to outgoing connections. One drawback of this benchmark is that it fails to account for the heterogeneity in complex networks. Another is that it does not allow embedded communities to overlap. A few benchmarks have been proposed for testing overlapping community detection algorithms, all of which are special cases of the planted l-partition model [Condon and Karp 2001] just like GM.

Sawardecker [Sawardecker et al. 2009] proposed an extension of GM, in which the probability p_{ij} of an edge being present in the network is a non-decreasing function based solely on the set of co-memberships of nodes i and j. With the definition $p_{ij} = p_k$, parameter p_k is the connection probability of nodes i and j that co-occur k times, subject to $p_0 < p_1 \le p_2 \le \cdots$.

The LFR[15] benchmark proposed in [Lancichinetti and Fortunato 2009a] introduces 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 $O_m \ge 1$ communities. The generating procedure is equivalent to generating a bipartite network where the two classes are the communities and nodes subject to the requirement that the sum of community sizes equals the sum of node memberships. LFR also provides a rich set of parameters to control the network topology, including the mixing parameter μ, the average degree $\overline{k}$, the maximum degree k_{max}, the maximum community size c_{max}, and the minimum community size c_{min}.

The LFR model brings benchmarks closer to the features observed in real-world networks. However, requiring that overlapping nodes interact with the same number of embedded communities is unrealistic in practice. A simple generalization, where each overlapping node may belong to different number of communities has been considered in [McDaid and Hurley 2010].

In [Gregory 2011], crisp communities from LFR are converted to fuzzy associations by adding a belonging coefficient to the occurrence of nodes. This coefficient can be defined as

$$p_{ij} = s_{ij} p_1 + (1 - s_{ij}) p_0,$$

where p_k is the same as in Sawardecker's model and $s_{ij} = \sum_{c \in C} \alpha_{ic} \alpha_{jc}$ is the similarity of node i and j as defined in (3). In other words, the probability of an edge being present depends not only on the number of communities in which nodes i and j appear together but also on their degree of belonging to these communities.

6. TESTS IN SYNTHETIC NETWORKS

In this section, we empirically compare the performance of different algorithms on LFR networks. We focus on algorithms which produce a crisp assignment of vertices to communities. In total, fourteen algorithms that we were able to collect and test are listed in Table I. Note that the time complexity given is for the worst case.

We use the default parameter setting for most of the algorithms if applicable. For LFM, we set α=1.0, which has previously been reported to give good results [Lancichinetti et al. 2009]. For $iLCD$, we set $fRatio = 0.5$ and $bThreshold = 0.3$. For other algorithms with tunable parameters, the results with the best setting

[15] http://sites.google.com/site/andrealancichinetti/files

are reported. For *Link*, the threshold varies from 0.1 to 0.9 with an interval 0.1. For COPRA, parameter v is taken from the range [1,10]. For SLPA, parameter r varies from 0.05 to 0.5 with an interval 0.05. Since COPRA and SLPA are non-deterministic, we repeated each of them 10 times on each network instantiation. For NMF, which returns a fuzzy assignment, we applied the same threshold as SLPA to convert it to a crisp assignment.

For each LFR network, we generated 10 instantiations. We used networks with sizes $n \in \{1000, 5000\}$. The average degree is kept at $\overline{k} = 10$, which is of the same order as most real-world social networks [16]. The rest of the parameters of LFR generator are set the same as in [29]: node degrees and community sizes are governed by power law distributions with exponents $\tau_1 = 2$ and $\tau_2 = 1$ respectively, the maximum degree is $k_{max} = 50$, and community sizes vary between $c_{min} = 20$ and $c_{max} = 100$. The mixing parameter μ is taken from the range $\{0.1, 0.3\}$, which is the expected fraction of links of a node connecting to other nodes in the same community.

The degree of overlap is determined by two parameters. O_n is the number of overlapping nodes, and O_m is the number of communities to which each overlapping node belongs. We fixed the former to be 10% of the total number of nodes. Instead of fixing the O_m, we allow it to vary from 2 to 8 indicating the diversity of overlapping nodes. By increasing the value of O_m, we create harder detection tasks in an intuitive way. This also allow us to look into more details of the detection accuracy at node level later on.

Two previously discussed measures, Omega and NMI, are used to quantify the quality of the cover discovered by an algorithm. In addition, to summarize the vast volume of comparison results and provide an intuitive measure of *relative* performance, we proposed $RS_M(i)$, the averaged ranking score for a given algorithm i with respect to some measure M as follows:

$$RS_M(i) = \sum_{j=1} w_j \cdot rank(i, O_m^j), \tag{12}$$

where O_m^j is the number of memberships in $\{2, 3, \cdots, 8\}$, w_j is the weight and function *rank* returns the ranking of algorithm i for the given O_m. For simplicity, we assume equal weights over different O_m's in this paper. Sorting RS_M in increasing order gives the final ranking among algorithms. Whenever it is clear from context, we use the term *ranking* to refer to the final ranking without the actual score value.

6.1 Uncovering Overlapping Communities in LFR

We first examine how the performance changes as the number of memberships O_m varies from small to large values (i.e., 2 to 8). The results for $n = 5000$ are shown in Figure 1 $\sim$ 4. More results for $n = 1000$ and detailed information of individual algorithms can be found in the Supplementary Information (SI).

In general, changes in the network topology, especially the mixing value μ, has a similar impact as previously shown in analyses of methods from disjoint community detection. That is, the larger the value of μ, the poorer the results produced by detection algorithms. However, increasing network size from 1000 to 5000, results

[16] snap.stanford.edu/data/

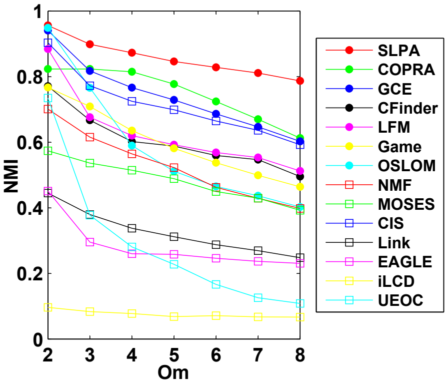


Fig. 1. NMI as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.1$.

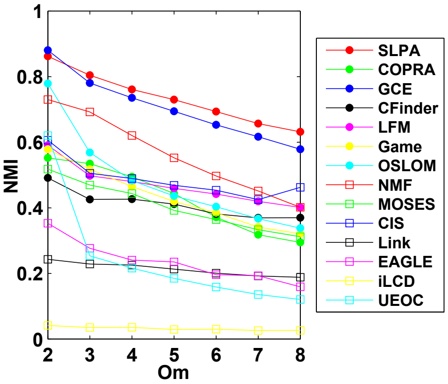


Fig. 2. NMI as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.3$.

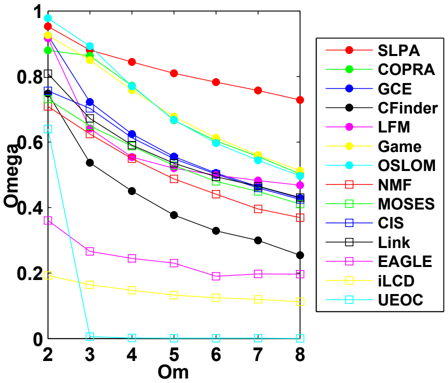


Fig. 3. Omega Index as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.1$.

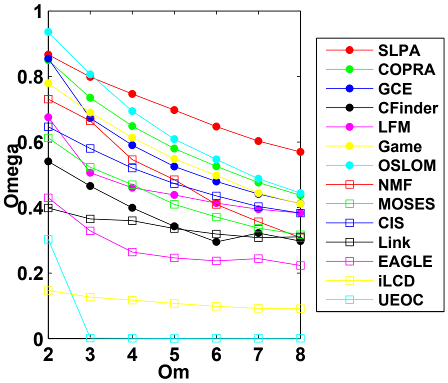


Fig. 4. Omega Index as a function of the number of memberships O_m for LFR networks with $n = 5000$ and $\mu = 0.3$.

in slightly better performance. On the other hand, performance decays as the degree of overlapping increases (i.e., O_m getting larger) for almost all algorithms. Most algorithms have a moderate rate of decay, except for OSLOM and UEOC as measured by NMI.

To simplify our analysis, we focus mainly on the case $n = 5000$ and $\mu = 0.3$, which is the hardest task in our tests. The information in Figure 2 and 4 is summarized by the rankings given in Table II, i.e., RS_{OMG} and RS_{NMI}. Although the rankings of algorithms are not consistent between the two measures, they share the same top eight algorithms, including SLPA, GCE, NMF, CIS, LFM, OSLOM, Game and COPRA. Three of them, SLPA, Game and COPRA, are agent based algorithms. Four others, GCE, CIS, OSLOM and LFM, are based on local expansion. NMF represents fuzzy clustering.

Table II. The ranking of algorithms.

Rank	RS_{OMG}	RS_{NMI}	RS_F
1	SLPA	SLPA	SLPA
2	OSLOM	GCE	CFinder
3	COPRA	NMF	Game
4	Game	CIS	OSLOM
5	GCE	OSLOM	COPRA
6	CIS	LFM	MOSES
7	NMF	COPRA	Link
8	LFM	Game	iLCD
9	MOSES	CFinder	LFM
10	CFinder	MOSES	UEOC
11	Link	EAGLE	CIS
12	EAGLE	UEOC	GCE
13	iLCD	Link	EAGLE
14	UEOC	iLCD	NMF

6.2 Comparing Detected Community Size Distribution in LFR

We compare the discovered and the known distribution of community sizes (CS) to verify the ranking we found above. Figure 5 $\sim$ 7 show the histogram of CS created from the information in Figure 2. In the synthetic networks, we expect the size distribution to follow the power law with exponent $\tau_2 = 1$ with the minimum size 20 and the maximum size 100. As shown in Figure 5, SLPA, GCE and NMF find a *unimodal* distribution with a single peak at $CS = 20$ that agrees well with the ground truth distribution. This explains why they perform well with respect to NMI. LFM and MOSES have peak around $CS = 5$, which lowers their performance. The prominent feature of Figure 6 (see the inset) is a *bimodal* distribution that have a peak at $CS = 1 \sim 5$. This means that algorithms like OSLOM, Game, COPRA, CIS and CFinder find significant number of small size communities. In Figure 7, the distribution is shifted mostly outside the predefined range 20$\sim$100. Algorithms with such a distribution create relatively small communities and perform poorly with respect to this analysis.

It is worth noticing that in Figure 6, outside the range with a peak, the distributions seem to agree well with the ground truth, especially for COPRA. In this range, the performances of OSLOM, Game, CIS and COPRA with respect to NMI are still fairly *stable*. This demonstrates that *NMI is to some degree not sensitive to small size communities (including outliers or singleton communities).*

6.3 Identifying Overlapping Nodes in LFR

Community overlap manifests itself as the existence of the nodes with membership in multiple communities. Thus, we will refer to nodes with multiple membership as overlapping nodes. In real-world social networks, such nodes are important because they usually represent bridges (or messengers) between communities. For this reason, the ability to identify overlapping *nodes*, although often *neglected*, is *essential* for assessing the accuracy of community detection algorithms. However, measures like NMI and Omega focus only on providing an *overall* measure of algorithmic accuracy. As we see in section 6.2, these measures might not be sensitive enough to provide an accurate picture of what is happening at the *node level*. In this section,

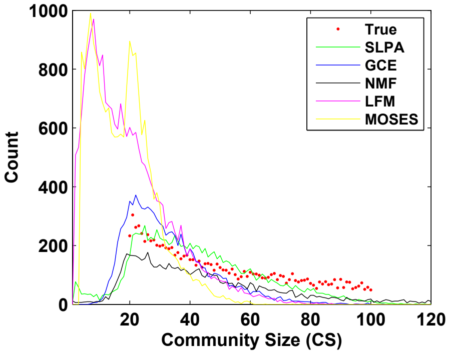


Fig. 5. Histogram of the detected community sizes for SLPA, GCE, NMF, LFM and MOSES created from the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

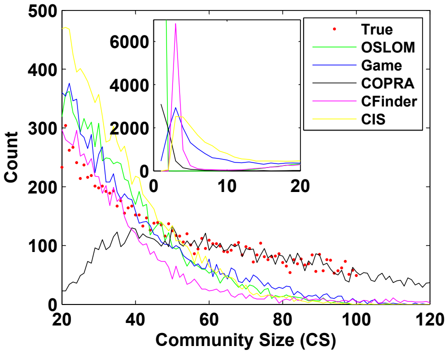


Fig. 6. Histogram of the detected community sizes for OSLOM, Game, COPRA, CIS and CFinder created from the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

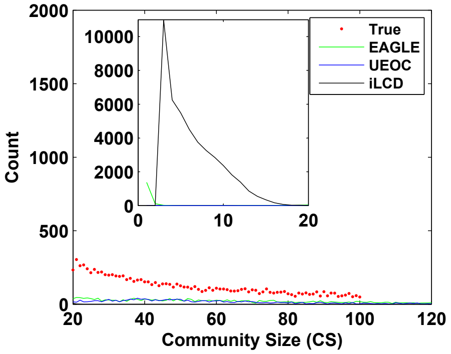
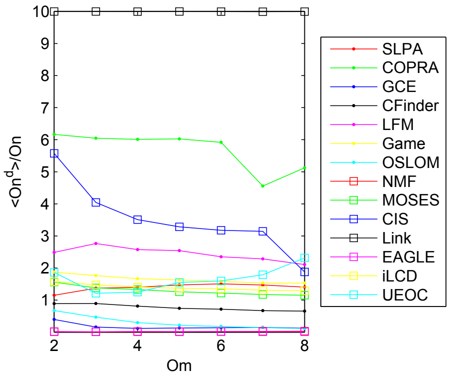


Fig. 7. Histogram of the detected community sizes for EAGLE, UEOC and iLCD crated from the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

we introduce a framework to evaluate an algorithm's ability to identify overlapping nodes.

We first introduce tests for the number of detected overlapping nodes (denoting as O_n^d) and detected memberships (denoting as O_m^d) based on the information in Figure 2. Figure 8 shows the number of overlapping nodes detected by various algorithms normalized by the true O_n embedded in the network. For the normalized number of memberships of the detected overlapping nodes, see Figure 9. These results imply that identifying a set of overlapping nodes with large number memberships is easier (with a value close to 1 on y-axis), while uncovering *all* the memberships of nodes with high O_m is *hard*.

Note that the number of overlapping nodes O_n^d alone is insufficient to accurately quantify the detection performance, because it contains both true and false positive. To provide more precise analysis, we consider the identification of overlapping nodes



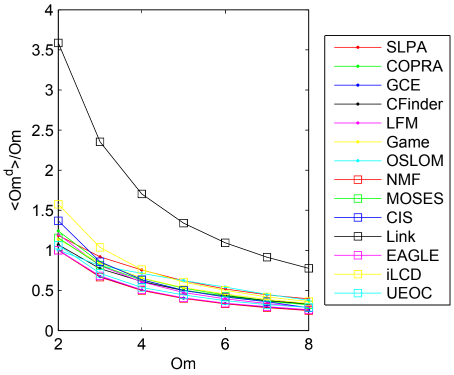


Fig. 8. The normalized number of detected overlapping nodes based on the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

Fig. 9. The normalized number of memberships of detected overlapping nodes based on the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

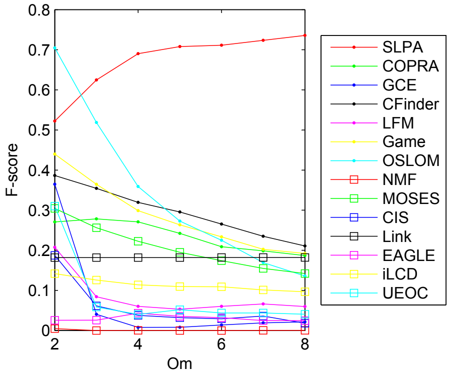


Fig. 10. The F-score as a function of the number of memberships based on the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

as a *binary classification* problem. A node is labeled as *overlapping* as long as $O_m{>}1$ or $O_m^d{>}1$ and labeled as *non-overlapping* otherwise. Within this framework, we can use F-score as a measure of detection accuracy defined as

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{13}$$

where *recall* is the number of correctly detected overlapping nodes divided by the expected value of overlapping nodes O_n, and *precision* is the number of correctly detected overlapping nodes divided by the total number of detected overlapping nodes O_n^d. F-score reaches its best and worst value at 1 and 0, respectively.

Results are shown in Figure 10, where the F-score decays as O_m increases except for SLPA. SLPA has a positive correlation with O_m while other algorithms typically demonstrate a negative correlation. This indicates that SLPA is able to *correctly* uncover *reasonable amount* of overlapping nodes. Algorithms such as COPRA,

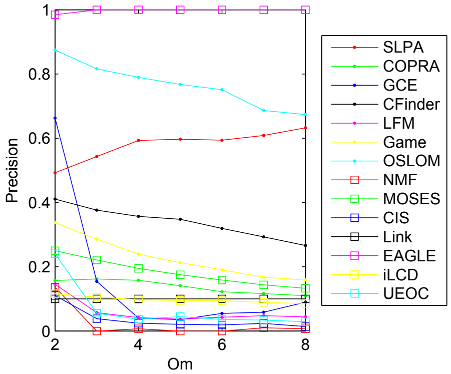


Fig. 11. The precision as a function of the number of memberships based on the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

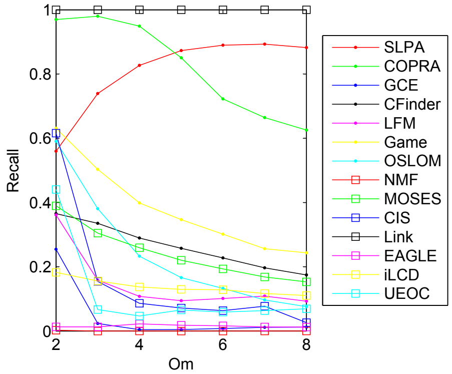


Fig. 12. The recall as a function of the number of memberships based on the results for LFR networks with $n = 5000$ and $\mu = 0.3$.

EAGLE, Link and OSLOM that fail to have a good balance between precision and recall result in lower F-score.

The ranking with respect to F-score, RS_F, is also shown in Table II. It is clear that the community quality ranking (e.g., RS_{NMI}) and node quality ranking (e.g., RS_F) provide quite different pictures of the performance. Algorithms with a low ranking with respect to RS_{NMI}, could actually have good performances when it comes to identifying overlapping nodes (e.g., iLCD, CFinder and MOSES), while high-ranking algorithms, including COPRA and LFM, might identify too many nodes as overlapping. Meanwhile, high-ranking algorithms could also under claim overlapping like NMF. In fact, the NMF with threshold 0.4 that gives the best NMI is not able to find overlapping nodes at all. It discovers merely disjoint communities. This suggests *the need for a careful treatment of the algorithms with a high NMI score* if the application of these algorithms is aimed at identifying nodes with multiple community memberships.

6.4 Final Ranking

In summary, with the F-score ranking, GCE, CIS, LFM and NMF no longer rank in the top eight algorithms, since they fail to correctly identify overlapping nodes. In contrast, CFinder, MOSES and Link demonstrate the ability to correctly classify such nodes. The disagreement between two types of rankings again suggests that considering solely community level measures might be dangerous because it may not offer accurate insight into the *overlapping* of interest. The node level quality tests are necessary, especially when dealing with sparse networks where the fraction of overlapping nodes is relatively small (as in our tests with an overlap rate of 10%). These two types of rankings provide complementary information, and which one is appropriate depends on application.

Taking both RS_{NMI} and RS_F into account (i.e., top eight in both), we conclude that SLPA, OSLOM, Game and COPRA offer better performance than the other tested algorithms relative to the identification and classification tasks. In particular,

Table III. Social networks in the tests

Network	n	$\bar{k}$	Network	n	$\bar{k}$
karate (KR)	34	4.5	PGP	10680	4.5
football (FB)	115	10.6	Email (EM)	33696	10.7
lesmis (LS)	77	6.6	P2P	62561	2.4
dolphins (DP)	62	5.1	Epinions (EP)	75877	10.6
CA-GrQc (CA)	4730	5.6	Amazon (AM)	262111	6.8

SLPA performs well and is remarkably stable over different ranking criteria and quality measures.

7. TESTS IN REAL-WORLD SOCIAL NETWORKS

In this section, we tested overlapping community detection algorithms in a wide range of social networks listed in Table III. More information about these networks can be found here[17].

We selected two overlapping modularities Q_{ov}^{E} in (8) and Q_{ov}^{Ni} in (11) as quality measures. The former is based on the node belonging factor, and the later is based on the link belonging factor. For the arbitrary function in Q_{ov}^{Ni}, we adopted the one used in [Gregory 2010], $f(x) = 60x - 30$. In Figure 13 $\sim$ 18, networks are shown in the order of increasing number of edges along the x-axis. Lines connecting points are meant merely to aid the reader in differentiating points from the same algorithm. We removed CFinder, EAGLE and NMF from the test due to either their memory or computation inefficiency in large networks. As a reference, we also performed disjoint community detection with the Infomap algorithm [M. Rosvall 2008], which has been shown to be quite accurate previously [Lancichinetti and Fortunato 2009b].

Figure 13 and Figure 14 show a positive correlation between the two quality measures. Typically, the disjoint partitioning achieves higher Q_{ov}^{E} than overlapping clusterings, which empirically serves as a bound of the quality of detected overlapping communities. This also holds for Q_{ov}^{Ni} in general.

In general, Link and iLCD achieve lower Q_{ov}^{Ni} or Q_{ov}^{E} compared to others, while SLPA, LFM, COPRA, OSLOM and GCE achieve higher performance on larger networks (e.g., last five networks). Moreover, an algorithms may not perform equally well on different types of network structures. Some of them are sensitive to specific structures. For example, only SLPA, LFM, CIS and Game have satisfying performances in networks with highly sparse structure such as $P2P$, for which COPRA finds merely one single giant community and GCE also fails. Another issue is that some algorithms tend to over-detect the overlap, as was the case for LFR

[17] CA-GrQc: a co-authorship network based on papers in General Relativity publishing in Arxiv.
PGP: a network of users of the Pretty-Good-Privacy algorithm.
Email: a communication network in Enron via emails [Leskovec et al. 2009].
Epinions: a who-trust-whom on-line social network of a consumer review site Epinions.com [Richardson et al. 2003].
P2P: the Gnutella peer-to-peer file sharing network from August 2002.
Amazon: a co-purchase network of the Amazon website [Leskovec et al. 2007].
Data are available at `http://www-personal.umich.edu/~mejn/netdata/` and `http://snap.stanford.edu/data/`

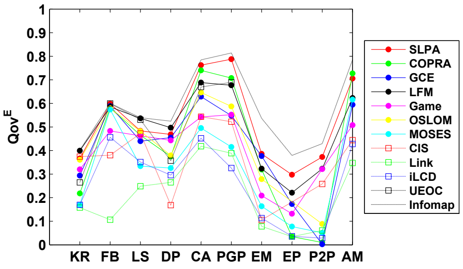


Fig. 13. Overlapping modularity Q_{ov}^{E} for social networks.

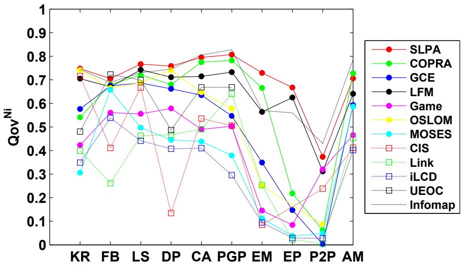


Fig. 14. Overlapping modularity Q_{ov}^{Ni} for social networks.

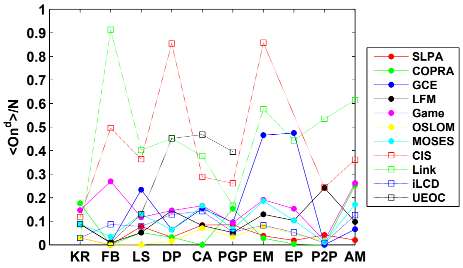


Fig. 15. The normalized number of detected overlapping nodes for social networks based on the clustering with the best Q_{ov}^{E}.

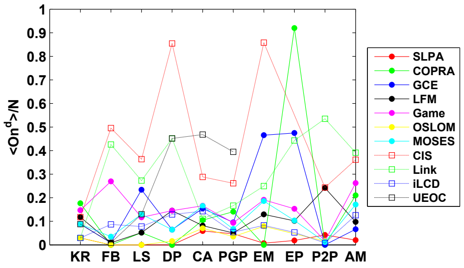


Fig. 16. The normalized number of detected overlapping nodes for social networks based on the clustering with the best Q_{ov}^{Ni}.

networks. CIS and Link fail in the test because they find too many overlapping nodes or memberships relative to the consensus shown by the other algorithms as seen in Figure 15 $\sim$ 18. Such over-detection happens to other algorithms, including COPRA, GCE and UEOC on specific networks, resulting in low performance for these algorithms.

Some interesting common features are observed from our tests, which are a relatively agreement between results of different algorithms. As shown in Figure 15 and 16, the fraction of overlapping nodes found by most of the algorithms is typically less than 30%. Results from SLPA, OSLOM and COPRA, which offer good performances in the LFR benchmarks, show an even smaller fraction of overlapping nodes, less than 20%, in most real-world networks examined in this paper. Moreover, Figure 17 and 18 confirm that the diversity (i.e., membership) of overlapping nodes in the tested social networks is relatively small as well, typically 2 or 3 .

8. CONCLUSIONS AND DISCUSSIONS

In this paper, we review a wide rang of overlapping community detection algorithms, various quality measures and several existing benchmarks. A number of tests are performed on the LFR benchmarks, incorporating different network structures and various degree of overlapping. We focus on crisp community detection. Quality evaluation is performed on both community and node levels to provide

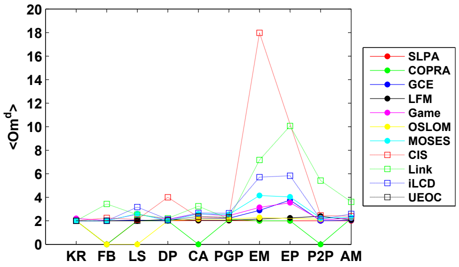


Fig. 17. The number of detected memberships for social networks based on the clustering with the best Q_{ov}^{E}.

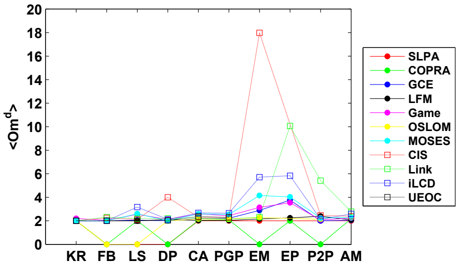


Fig. 18. The number of detected memberships for social networks based on the clustering with the best Q_{ov}^{Ni}.

complementary information. The proposed node level evaluation framework helps to reveal the problems of *overdetection* or *underdetection* which call for attention when designing or evaluating detection algorithms. The results discovered in real-world social networks suggest the sensitivity of some algorithms to sparse networks. A common feature of social networks in view of agreement of different algorithms is relatively small number of overlapping nodes, most of which belong just to a few communities.

Despite the large amount of work devoted to developing detection algorithms, there are a number of fundamental questions that have yet to be fully addressed. Two of the most prominent involve *when to apply overlapping methods* and *how significant the overlapping is.*

It is natural to ask whether or not the application of the overlapping detection algorithms captures any additional information that a disjoint algorithm would necessarily miss. Unfortunately, measures like NMI and Omega do not offer a satisfying answer. The discussion on the necessity of overlap has largely been unexplored. In [Kelley 2009], the authors empirically examined attributes of the vertices in a network representing commenting activity. The authors suggest that, for a pair of communities A and B, the trait similarity between $A \cap B$ and the sets $A - B$ and $B - A$ be higher than the similarity between $A - B$ and $B - A$. Such a relationship might offer a way to estimate the validation of the overlap.

The identification of the significance of community structures has been previously explored only within the context of disjoint community detection and following the notion of modularity [Reichardt and Bornholdt 2006], [Guimerà et al. 2004], [Massen and Doye 2005]. The robustness and uniqueness of a discovered partitioning is also examined in [Gfeller et al. 2005; Karrer et al. 2008; Massen and Doye 2007]. Many of these techniques can be extend to assess the overlapping community structure. Interestingly, statistical significance has begun to work itself into detection methodologies such as OSLOM [Lancichinetti et al. 2011].

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: `http://www.acm.org/pubs/citations/journals/tocl/2011-V-N/p1-URLend`.

ACKNOWLEDGMENTS

This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and by the Office of Naval Research Grant No. N00014-09-1-0607. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied of the Army Research Laboratory, the Office of Naval Research, or the U.S. Government.

The authors would like to thank researchers who generously provided their software and help setting up our experiments. We especially thank Steve Gregory for his helpful codes and discussions.

REFERENCES

Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. 2010. Link communities reveal multiscale complexity in networks. *Nature 466*, 761–764.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. 1999. Optics: ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*. 49–60.

Arenas, A., Díaz-Guilera, A., and Pérez-Vicente, C. J. 2006. Synchronization reveals topological scales in complex networks. *Phys. Rev. Lett. 96,* 11, 114102.

Baumes, J., Goldberg, M., Krishnamoorthy, M., Magdon-Ismail, M., and Preston, N. 2005. Finding communities by clustering a graph into overlapping subgraphs. In *IADIS*.

Baumes, J., Goldberg, M., and Magdon-Ismail, M. 2005. Efficient identification of overlapping communities. *LNCS 3495*, 27–36.

Bianconi, G., Pin, P., and Marsili, M. 2008. Assessing the relevance of node features for network structure. *Proceedings of the National Academy of Sciences of the United States of America* 28, 7.

Blatt, M., Wiseman, S., and Domany, E. 1996. Superparamagnetic clustering of data. *Phys. Rev. Lett. 76*, 3251–3254.

Breve, F., Zhao, L., and Quiles, M. 2009. Uncovering overlap community structure in complex networks using particle competition. In *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*. 619–628.

Campello, R. J. G. B. 2007. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recogn. Lett. 28*, 833–841.

Campello., R. J. G. B. 2010. Generalized external indexes for comparing data partitions with overlapping categories. *Pattern Recogn. Lett. 31*, 966–975.

Cazabet, R., Amblard, F., and Hanachi, C. 2010. Detection of overlapping communities in dynamical social networks. In *SOCIALCOM*. 309–314.

Chen, D., Shang, M., Lv, Z., and Fu, Y. 2010. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications 389,* 19, 4177–4187.

Chen, J., Zaïane, O. R., and Goebel, R. 2009. A visual data mining approach to find overlapping communities in networks. In *Advances in Social Networks Analysis and Mining*. 338–343.

Chen, W., Liu, Z., Sun, X., and Wang, Y. 2010. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov. 21*, 224–240.

Collins, L. M. and Dent, C. W. 1988. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research 23,* 2, 231–242.

Condon, A. and Karp, R. M. 2001. Algorithms for graph partitioning on the planted bisection model. *Random Structures and Algorithms 18*, 116–140.

Danon, L., Duch, J., Arenas, A., and Diaz-guilera, A. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 09008.

Davis, G. B. and Carley, K. 2008. Clearing the fog: Fuzzy, overlapping groups for social networks. *Social Networks 30,* 3, 201–212.

Ding, F., Luo, Z., Shi, J., and Fang, X. 2010. Overlapping community detection by kernel-based fuzzy affinity propagation. In *Intelligent Systems and Applications*.

Du, N., Wang, B., and Wu, B. 2008. Overlapping community structure detection in networks. In *Proceeding of the 17th ACM conference on Information and knowledge management*. 1371–1372.

Evans, T. 2010. Clique graphs and overlapping communities (arxiv: 1009.0638).

Evans, T. and Lambiotte, R. 2010. Line graphs of weighted networks for overlapping communities. *Eur. Phys. J. B 77*, 265.

Evans, T. S. and Lambiotte, R. 2009. Line graphs, link partitions and overlapping communities. *Phys. Rev. E 80*, 016105.

Farkas, I., Ábel, D., Palla, G., and Vicsek, T. 2007. Weighted network modules. *New Journal of Physics 9,* 6, 180.

Fisher, D. C. 1989. Lower bounds on the number of triangles in a graph. *Journal of Graph Theory 13,* 4, 505–512.

Fortunato, S. 2010. Community detection in graphs. *Physics Reports 486*, 75–174.

Fortunato, S. and Barthelemy, M. 2007. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA 104*, 36–41.

Frey, B. J. and Dueck, D. 2007. Clustering by passing messages between data points. *Science 315*, 972–976.

Fu, Q. and Banerjee, A. 2008. Multiplicative mixture models for overlapping clustering. In *ICDM '08*. 791 –796.

Geweniger, T., Zühlke, D., Hammer, B., and Villmann, T. 2009. Fuzzy variant of affinity propagation in comparison to median fuzzy c-means. In *Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps*. 72–79.

Gfeller, D., Chappelier, J.-C., and De Los Rios, P. 2005. Finding instabilities in the community structure of complex networks. *Phys. Rev. E 72*, 056135.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proc Natl Acad Sci USA 99,* 12, 7821–7826.

Gregory, S. 2007. An algorithm to find overlapping community structure in networks. *Lect. Notes Comput.Sci.*.

Gregory, S. 2008. A fast algorithm to find overlapping communities in networks. *Lect. Notes Comput. Sci. 5211*, 408.

Gregory, S. 2009. Finding overlapping communities using disjoint community detection algorithms. *Complex Networks 207*, 47–61.

Gregory, S. 2010. Finding overlapping communities in networks by label propagation. *New J. Phys. 12*, 10301.

Gregory, S. 2011. Fuzzy overlapping communities in networks. *Journal of Statistical Mechanics: Theory and Experiment 2011,* 02, P02017.

Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. 2004. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E 70*, 025101.

Havemann, F., Heinz, M., Struck, A., and Glaser, J. 2011. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *Journal of Statistical Mechanics: Theory and Experiment 2011,* 01, P01023.

Hubert, L. and Arabie, P. 1985. Comparing partitions. *Journal of Classification 2*, 193–218.

Hüllermeier, E. and Rifqi, M. 2009. A fuzzy variant of the rand index for comparing clustering structures. In *IFSA/EUSFLAT Conf.* 1294–1298.

Jin, D., Yang, B., Baquero, C., Liu, D., He, D., and Liu, J. 2011. A markov random walk under constraint for discovering overlapping communities in complex networks. *Journal of Statistical Mechanics: Theory and Experiment 2011,* 05, P05031.

Karrer, B., Levina, E., and Newman, M. E. J. 2008. Robustness of community structure in networks. *Phys. Rev. E 77*, 046119.

Kelley, S. 2009. The existence and discovery of overlapping communities in large-scale networks. Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY.

Kelley, S., Goldberg, M., Magdon-Ismail, M., Mertsalov, K., and Wallace, A. 2011. *Handbook of Optimization in Complex Networks.* Springer, Chapter 6.

Kim, Y. and Jeong, H. 2011. The map equation for link community (unpublished).

Kovács, I. A., Palotai, R., Szalay, M., and Csermely, P. 2010. Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS ONE 5,* 9, e12528.

Kumpula, J. M., Kivelä, M., Kaski, K., and Saramäki, J. 2008. Sequential algorithm for fast clique percolation. *Phys. Rev. E 78,* 2, 026109.

Lancichinetti, A. and Fortunato, S. 2009a. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E 80,* 1, 016118.

Lancichinetti, A. and Fortunato, S. 2009b. Community detection algorithms: a comparative analysis. *Phys. Rev. E 80*, 056117.

Lancichinetti, A., Fortunato, S., and Kertesz, J. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys. 11*, 033015.

Lancichinetti, A., Fortunato, S., and Kertész, J. 2009. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics 11*, 033015.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. 2011. Finding statistically significant communities in networks. *PLoS ONE 6,* 4, e18961.

Latouche, P., Birmele, E., and Ambroise, C. 2011. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics 5*, 309C336.

Lázár, A., Ábel, D., and Vicsek, T. 2010. Modularity measure of networks with overlapping communities. *Europhysics Letters 90,* 1, 18001.

Lee, C., Reid, F., McDaid, A., and Hurley, N. 2010. Detecting highly overlapping community structure by greedy clique expansion. In *Proc. 4th Int. Workshop on Social Network Mining and Analysis.* 33–42.

Leskovec, J., Adamic, L. A., and Huberman, B. A. 2007. The dynamics of viral marketing. *ACM Trans. Web 1.*

Leskovec, J., Lang, K. J., and andMichael W. Mahoney, A. D. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics 6*, 29–123.

Leskovec, J., Lang, K. J., and Mahoney, M. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web.* 631–640.

Li, D., Leyva, I., Almendral, J., Sendina-Nadal, I., Buldu, J., Havlin, S., and Boccaletti, S. 2008. Synchronization interfaces and overlapping communities in complex networks. *Phys. Rev. Lett. 101*, 168701.

Lu, Q., Korniss, G., and Szymanski, B. K. 2009. The naming game in social networks:community formation and consensus engineering. *Journal of Economic Interaction and Coordination 4,* 2, 221–235.

M. Rosvall, C. B. 2008. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. 105*, 1118–1123.

Magdon-ismail, M. and Purnell, J. 2011. Fast overlapping clustering of networks using sampled spectral distance embedding and gmms. Tech. rep., Rensselaer Polytechnic Institute.

Massen, C. and Doye, J. 2005. Identifying communities within energy landscapes. *Phys. Rev. E 71*, 046101.

Massen, C. and Doye, J. 2007. Thermodynamics of community structure. *Preprint arXiv:cond-mat/0610077v1*.

McDaid, A. and Hurley, N. 2010. Detecting highly overlapping communities with model-based overlapping seed expansion. In *Advances in Social Networks Analysis and Mining.* 112 –119.

MOLLOY, M. AND REED, B. 1995. A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms 6*, 161–179.

MOON, J. AND MOSER, L. 1965. On cliques in graphs. *Israel Journal of Mathematics 3*, 23–28.

NEPUSZ, T., PETRÓCZI, A., NÉGYESSY, L., AND BAZSÓ, F. 2008. Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E 77*, 016107.

NEWMAN, M. E. J. 2004. Fast algorithm for detecting community structure in networks. *Phys. Rev. E 69*, 066133.

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

NEWMAN, M. E. J. AND LEICHT, E. A. 2007. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences 104*, 9564–9569.

NEWMAN, M. E. J., STROGATZ, S. H., AND WATTS, D. J. 2001. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E 64,* 2, 026118.

NICOSIA, V., MANGIONI, G., CARCHIOLO, V., AND MALGERI, M. 2009. Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech.*, 03024.

NOWICKI, K. AND SNIJDERS, T. A. B. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association 96,* 455, 1077–1087.

PADROL-SUREDA, A., PERARNAU-LLOBET, G., PFEIFLE, J., AND MUNTS-MULERO, V. 2010. Overlapping community search for social networks. In *ICDE'10*. 992–995.

PALLA, G., DERÉNYI, I., FARKAS, I., AND VICSEK, T. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature 435*, 814–818.

PSORAKIS, I., ROBERTS, S., EBDEN, M., AND SHELDON, B. 2011. Overlapping community detection using bayesian non-negative matrix factorization. *Phys. Rev. E 83,* 6, 066114.

RAGHAVAN, U. N., ALBERT, R., AND KUMARA, S. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E 76*, 036106.

REES, B. AND GALLAGHER, K. 2010. Overlapping community detection by collective friendship group inference. In *Advances in Social Networks Analysis and Mining*. 375 –379.

REICHARDT, J. AND BORNHOLDT, S. 2004. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett. 93*, 218701.

REICHARDT., J. AND BORNHOLDT, S. 2006. Statistical mechanics of community detection. *Phys. Rev. E 74,* 1, 016110.

REICHARDT, J. AND BORNHOLDT, S. 2006. When are networks truly modular? *Physica D Nonlinear Phenomena 224*, 20–26.

REN, W., YAN, G., LIAO, X., AND XIAO, L. 2009. Simple probabilistic algorithm for detecting community structure. *Phys. Rev. E 79,* 3, 036111.

RICHARDSON, M., AGRAWAL, R., AND DOMINGOS, P. 2003. Trust management for the semantic web. In *ISWC 2003*. Vol. 2870. 351–368.

SAWARDECKER, E., SALES-PARDO, M., AND AMARAL, L. 2009. Detection of node group membership in networks with group overla. *Eur. Phys. J. B 67*, 277.

SHEN, H., CHENG, X., CAI, K., AND HU, M.-B. 2009. Detect overlapping and hierarchical community structure. *Physica A 388*, 1706.

SHEN, H., CHENG, X., AND GUO, J. 2009. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics: Theory and Experiment* 07, 9.

WANG, X., JIAO, L., AND WU, J. 2009. Adjusting from disjoint to overlapping community detection of complex networks. *Physica A 388*, 5045–5056.

WHITE, S. AND SMYTH, P. 2005. A spectral clustering approach to finding communities in graphs. In *Proc of SIAM International Conference on Data Mining*. 76–84.

WU, Z., LIN, Y., WAN, H., AND TIAN, S. 2010. A fast and reasonable method for community detection with adjustable extent of overlapping. In *2010 International Conference on Intelligent Systems and Knowledge Engineering*.

XIE, J. AND SZYMANSKI, B. K. 2011. Community detection using a neighborhood strength driven label propagation algorithm. In *IEEE NSW 2011*. 188–195.

Xie, J., Szymanski, B. K., and Liu, X. 2011. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *IEEE ICDM 2011 Workshop on DMCCI*.

Zarei, M., Izadi, D., and Samani, K. A. 2009. Detecting overlapping community structure of networks based on vertex-vertex correlations. *Journal of Statistical Mechanics: Theory and Experiment 2009,* 11, P11013.

Zhang, S., Wang, R.-S., and Zhang, X.-S. 2007. Uncovering fuzzy community structure in complex networks. *Phys. Rev. E 76,* 4, 046103.

Zhang, S., Wangb, R.-S., and Zhang, X.-S. 2007. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A 374*, 483–490.

Zhang, Y., Wang, J., Wang, Y., and Zhou, L. 2009. Parallel community detection on large networks with propinquity dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 997–1006.

Zhao, K., Zhang, S.-W., and Pan, Q. 2010. Fuzzy analysis for overlapping community structure of complex network. In *Control and Decision Conference, 2010 Chinese*. 3976 –3981.

Received Month Year; revised Month Year; accepted Month Year