Expert Systems with Applications 41 (2014) 4309–4321

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient algorithm for community mining with overlap in social networks

Delel Rhouma [a,*], Lotfi Ben Romdhane [b]

[a] MARS (Modeling of Automated Reasoning Systems) Research Group, FSM/University of Monastir, Tunisia
[b] Insitute of Computer Science and Telecom (ISITCom), University of Sousse, Tunisia

ARTICLE INFO

Keywords:
Social networks
Communities
Overlap
Objective function
Fuzzy membership degree

ABSTRACT

Detecting communities in social networks represents a significant task in understanding the structures and functions of networks. Several methods are developed to detect disjoint partitions. However, in real graphs vertices are often shared between communities, hence the notion of overlap. The study of this case has attracted, recently, an increasing attention and many algorithms have been designed to solve it. In this paper, we propose an overlapping communities detecting algorithm called DOCNet (Detecting overlapping communities in Networks). The main strategy of this algorithm is to find an initial core and add suitable nodes to expand it until a stopping criterion is met. Experimental results on real-world social networks and computer-generated artificial graphs demonstrate that DOCNet is efficient and highly reliable for detecting overlapping groups, compared with four newly known proposals.

1. Introduction

Graphs play a central role in the field of complex systems. Indeed they are the preferred tool for mathematical modeling. We find it naturally in the study of several areas: sociology, biology, linguistics, physics, computer science ... (Pons, 2004). These graphs can reach large sizes and in more than hundred nodes, it becomes difficult to understand their structures and to view it legibly (Pons, 2004). The search for strongly linked groups of vertices can provide a simplified representation of the structure of large graphs: this fact is important for the end user because it allows to understand very intuitively the modeled social network. Thus, it brings the members of network by affinity or common characteristics, it is enrolled in the context of community detection. This represents one of the key problems in social network analysis and it has been extensively studied (Pons, 2004). These studies are divided into two families: finding homogenous communities (Fortunato, 2010; Zardi & Ben Romdhane, 2013; Cheong, Huynh, Lo, & Goh, 2013) or extracting a set of pairs of communities that behave in opposite ways with one another (exhibiting antagonistic behaviors) (Zhang, Lo, Lim, & Prasetyo, 2013; Lo, Surian, Prasetyo, Zhang, & Lim, 2013).

However, we should notice that in most of existing approaches the computed partitions are disjoint; i.e., each vertex is assigned to a single community. However, it is well understood that people in a social network are naturally characterized by multiple community memberships, hence the notion of overlap between communities. For example, a person usually has connections to several social groups like family, friends and colleagues. He can be an active member simultaneously in the fields of mathematics, biology, science, etc. Another typical example is in the PPI networks (protein–protein interaction) (Fortunato, 2010) in which we seek to identify functional classes. Indeed, many proteins have multiple functions depending on different tissues. They may belong to more than one functional unit and sometimes they act as a bridge that allows the transfer of information. So the assignment of this gene to a single class is not justifiable. For this reason, overlapping community detection algorithms have been investigated (Xie, Kelley, & Szymanski, 2013).

In this paper, we propose an efficient algorithm to identify overlapping nodes. It is based on the local optimization of a fitness function and a fuzzy belonging degree of different nodes. This membership is not only based on the number of link which connects the node to the community, but also on the size of the community and the shortest path from the node to all its members. We propose an objective function to qualify the overall quality of a partition; and present DOCNet, an algorithm for its optimization. The rest of this paper is organized as follows. In Section 2, we review related work; while in Section 3 we introduce preliminary material. Section 4 outlines the fundamentals of our model. In Section 5, we report experimental results and the final section offers concluding remarks and sheds light on future research directions.

2. Related works

Detecting overlapping communities is a task of a great importance in our world. Indeed, it is treated by several approaches which are reviewed and categorized into five classes that reflect how communities are identified (Xie et al., 2013).

* Corresponding author. Tel.: +216 95574879.
E-mail addresses: rh.delel@gmail.com (D. Rhouma), lotfi.ben.romdhane@usherbrooke.ca (L.B. Romdhane).

0957-4174/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2014.01.002

2.1. Graph theory partition

The first family is based on graph theory and the most popular technique in this approach is the Clique Percolation Method (CPM) proposed by Palla, Derenyi, Farkas, and Vicse (2005). It is based on the concept that the internal links in a community are likely to form cliques due to their high density. The main idea of this method is to move a clique on a graph, in some way, so it would probably be trapped inside its original community because it could not cross the bottleneck formed by the inter-community links. CPM is suitable for networks with dense connected parts. However K-clique cannot reach vertices with degree one (“leaves”) (Fortunato, 2010). In addition, it is very costly (Palla et al., 2005).

2.2. Link partition

Another line of research is link partition (Evans & Lambiotte, 2009; Ahn, Bagrow, & Lehmann, 2009; Wu, Lin, Wan, & Tian, 2010; Kim & Jeong, 2011). Indeed, it may happen that communities are joined to each other through their overlapped nodes without an inter-cluster edge. So it has been recently suggested to define community as sets of edges (Xie et al., 2013). The basic idea of Evans's method (Evans & Lambiotte, 2009) is to transform the original graph to a line graph i.e., each vertex in the line graph corresponds to an original edge and a link in the line graph represents the adjacency between two edges in the original graph. Nevertheless, this algorithm is memory inefficient (Tang, Wang, & Liu, 2012), so it cannot be applied to large social networks. Ahn and Al (Ahn et al., 2009) suggested an hierarchical clustering of links and computed the similarity between two links using Jaccard Index. The time complexity of this algorithm is $O\left(nk_{max}^2\right)$, where k_{max} is the maximum degree of node and $n$is the number of vertices in the network.

2.3. Local expansion and optimization partition

The idea of growing a partial community has also been explored (Xie et al., 2013). It relies on a fitness function characterizing the local quality of dense groups of nodes. Different overlapped groups can be locally optimal, so the vertices can be shared between communities. Baumes, Goldberg, Krishnamoorthy, Magdon-Ismail, and Preston (2005) proposed the iterative scan algorithm (IS) which starts with a candidate and adds or removes vertex as long as the function related to the density of link strictly improves. LFM (Lancichinetti, Fortunato, & KertTsz, 2009) develops a community from a random starting node until the objective function is not maximized. This method depends on a parameter that controls the size of formed groups. EAGLE (Shen, Cheng, Cai, & Hu, 2008) uses the agglomerative framework to produce a dendrogram. First, all maximal cliques are found and considered as first communities. Then the pair of communities with maximum similarity are merged. The optimal cut in the dendrogram is determined by the modularity. EAGLE is computationally expensive with complexity $O(n^2 + (h+n)s)$ (Xie et al., 2013), where s is the number of maximal cliques and h is the number of pairs of maximal cliques which are neighbors. GCE (Lee, Reid, McDaid, & Hurley, 2010) identifies cliques as seeds and expands them in greedy way. GCE also removes the communities that are similar using a function which computes the distance between communities. OSLOM (Lancichinetti, 2011) which is a multi-purpose technique, tests the statistical significance of a cluster with respect to a global null model during community expansion. Its main idea is to progressively add and remove vertices within the community so that to improve its fitness function. This process is repeated several times starting from different nodes in order to explore different regions of the graph. Its time complexity is $O(n^2)$. This family neglects communities of small sizes. An improvement of GCE and OSLOM is given by its conjunction, with WERW-Kpath (Fiumara, De Meo, Ferrara, & Provetti, 2013) algorithm which is a preprocessing step in which edges are weighted according to their centrality. This algorithm enhances the modularity and the quality of the community structure of these methods.

2.4. Fuzzy partition

The fourth approach is based on fuzzy clustering (Xie et al., 2013). It quantifies the strength of association between all nodes and communities and determines its adhesion to a group or not according to this degree. The most famous algorithm in this class is FCM which minimizes the intra-cluster variance by reducing its objective function (Bezdek, Ehrlich, & Full, 1984). FCM loses the graph structure because it takes into account only the distances between nodes. Nepusz, Petrczi, Ngyessy, and Bazs (2008) modeled the overlapping community detection as a nonlinear constrained optimization problem which can be solved by simulated annealing methods. NMF (Psorakis, Roberts, & Ebden, 2011) is a model based on Bayesian nonnegative matrix factorization. We may cite also OSBM (Latouche, Birmele, & Ambroise, 2011; Gregory, 2010), etc. Yet, these fuzzy approaches compute communities with spherical shapes mainly due to the constraints imposed on the membership degrees (Bezdek et al., 1984; Latouche et al., 2011; Psorakis et al., 2011). This is a major shortcoming since in real-networks, communities are of arbitrary shapes.

2.5. Agent-based partition

Finally, the Agent-based (Xie et al., 2013) approach uses labels to identify the membership of vertices and propagate it between neighbors, a node can have more than one label. In COPRA (Gregory, 2010), nodes update their belonging coefficients by averaging the coefficients from all its neighbors in a synchronous way. Its time complexity is $O(\log(\frac{vm}{n}))$ by iteration, where n is the number of vertices and m is the number of links and v is a parameter. SLPA (Xie, Szymanski, & Liu, 2011) spreads labels between nodes according to pairwise interaction rules and provides each node with a memory to store received information. Multi-state spin models (Reichardt & Born holdt, 2004) aim to minimize the equation of Hamiltonian. Despite, the high speed of this type of methods, they produce only small communities in some networks.

Despite the attempt of various methods to overcome the detection of overlapping communities, this problem still remains. Since it is an NP-hard problem (Fortunato, 2010) and some unstable nodes lying at the border between communities are often hard to classify into one community. Inspired by the above approaches, in this paper, a new Local Expansional algorithm called DOCNet, based on node fuzzy membership degree, proposed to detect the overlapping community structures. But, before going into its details, we need to introduce the following preliminary concepts.

3. Preliminaries

3.1. Problem formulation

We consider an undirected graph $G = (V, E)$, with $n = \mid V \mid$ nodes and $m = \mid E \mid$ edges. The purpose of the detection of overlapping communities in G is to determine a partition $P = \{C_1, \ldots, C_x\}$ of all the nodes of G where communities may be joined to each others (overlapped) ($\exists C_i \cap C_j \neq \emptyset,\ i \neq j$). A community may generally be described as group of nodes that probably share common properties and/or play similar roles within a network (Fortunato, 2010). It is a tight group with a high density of inter-community connections and a low density of intra-community connections (two communities can overlap since a node can belong to more than one).

This partition recovers all the nodes of G and do not require a priori knowledge of the number or the size of communities to be built.

3.2. Basic definitions

Definition 1 (*Cardinality of a community*). The cardinality of a community C is the number of its vertices. It is denoted by $|C|$.

Definition 2 (*Direct neighbor*). In the graph $G = (V, E)$, the vertex v is a direct neighbor of the node u if v and u are connected by an edge. This relationship is represented by the edge $(v, u) \in E$.

Definition 3 (*Vertex border*). It is all the direct neighbors of node v in the graph. This set is noted by $B(v)$. More formally this quantity is noted as follows:

$$B(v) = \{u \in V;\ \{u, v\} \in E\} \tag{1}$$

Definition 4 (*Internal Degree of a vertex to a community*). We call internal degree of a vertex v to a community C as the number of edges that point towards members of C. We note it $d_{in}(v, C)$.

$$d_{in}(v, C) = |\{(v, v') \in E,\ \ v' \in C\}| \tag{2}$$

Definition 5 (*External Degree of a vertex to a community*). We call external degree of a node v to a community C as the number of its direct neighbors who are not in C. We note it $d_{ext}(v, C)$.

$$d_{ext}(v, C) = |\{(v, v') \in E,\ \ v' \notin C\}| \tag{3}$$

Definition 6 (*Complete community*). In a complete community C, each pair of vertices is connected ($\forall v_i, v_j \in C;\ \exists (v_i, v_j) \in E$).

Definition 7 (*Overlap between two communities*). The overlap between two communities is the set of vertices shared between them.

$$Over(C_i, C_j) = \{v \mid v \in C_i \text{ and } v \in C_j\} \tag{4}$$

The more this quantity is bigger, the more we have a resemblance between both communities.

3.3. Fundamental concepts

3.3.1. Importance of a node

First, we should argue that the choice of random seeds where the community exploration starts may affect obtained covers. This means, in principle, that we cannot rely on random seeds (like other methods mentioned in the literature). We have found that, in a social network, a node whose neighbors are also connected (i.e., "know" one "another"), is the most relevant choice which tends to be the most compact core to form a group. That's why we have developed a new factor that measures the importance of an individual in a network that represents the starting point in the formation of a community. Indeed, for a network G with n nodes and m edges, the node importance of node u is its tendency to be the center of a community. This term combines two concepts:

1. The size of the border of node u (the degree of u) noted $|B(u)|$. This term is a function of first-order which takes into account the direct neighbors of vertex u.
2. The local coefficient of clustering of this node, which is a function of second order, takes into account the connections between neighboring vertices of u, and is defined by:

$$cfc(v) = \frac{2 \mid \{e_{jk}\} \mid}{\mid B(v) \mid (\mid B(v) \mid -1)};\quad v_j, v_k \in V,\ e_{jk} \in E \tag{5}$$

The coefficient of clustering was introduced by Watts and Strogatz (1998). This coefficient models how much a node and its neighbors can form a clique.

Definition 8 (*Node Importance*). Given a graph G and a node u, Node-Importance of u is a factor that measures how much this node is able to form a group. This index is noted NI and defined as:

$$NI(u) = cfc(u) * \mid B(u) \mid \tag{6}$$

where $cfc(u)$ is the clustering coefficient of u and $|B(u)|$ its border size. We remark that the importance of a node increases as its neighborhood increases and as these neighbors are also connected. Stated otherwise, the importance of a node increases as it becomes a "central influential" node in the network.

3.3.2. Membership degree

Our definition of the membership degree of a node v to a community C should not only take into consideration the interactions between v and C; but also the interactions between members of C. But, before outline of our idea, we need the subsequent definitions.

Definition 9 (*Distance between two nodes*). The distance between two nodes u and v of graph G, noted as $dist(u, v)$, is the number of edges being in the shortest path which leads u towards v. In a weighted graph, we add the weights of the links of the path.

Property 1. *For any node u and v from the graph G such that* $u \neq v$ *we have:* $0 < dist(u, v) \leqslant diam(G)$

Property 2. *Let v be a node belonging to a community C. If u is a node of the graph G as* $u \notin C$ *then we have:*

$$dist(u, v) \geqslant 1$$

Definition 10 (*Average distance between a node and a community*). It is the sum of distances of node u to different nodes $v \in C$, divided by the cardinality of C. It is given by:

$$dist_{moy}(u, C) = \begin{cases} \frac{\sum_{v \in C} dist(u,v)}{|C-1|} & \text{If } u \in C \\ \frac{\sum_{v \in C} dist(u,v)}{|C|} & \text{Otherwise} \end{cases} \tag{7}$$

Theorem 1. *For every node u of a graph G such that* $u \notin C$. *If* $dist_{moy}(u, C) = 1$ *then* $\{\{u\} \cup C\}$ *is a complete community (see Definition 6).*

Proof. Let $u \notin C$.

We have $dist_{moy}(u, C) = 1$ this is equivalent to say that $\frac{\sum_{v \in C} dist(u,v)}{|C|} = 1$ or even $\sum_{v \in C} dist(u, v) = |C|$.

However, according to the Property 2, $dist(u, v) \geqslant 1$. $|C|$ is a sum of distances $|C|$ each of which is of size greater than or equal to 1 then we necessarily have:

$$dist(u, v) = 1, \quad \forall v \in C$$

Thereafter there is a path having a distance of size 1 that leads from u to wholes nodes of C and hence the union of u with C is a complete community (see definition 6). □

Property 3. *For all nodes v of the graph Gwe have:* $0 < \frac{1}{dist_{moy}(v,C)} \leqslant 1$

Definition 11 (*Weighting coefficient*). It is the degree of compactness of one node u to a community C. More formally it is noted as follows:

$$\rho(u,C) = \frac{|B(u)|}{d_{in}(u,C)} \tag{8}$$

Property 4. *For all nodes u of the graph G we have:* $0 \leqslant \frac{1}{\rho(u,C)} \leqslant 1$

Now, we are ready to define the membership degree as follows.

Definition 12 (*Membership degree*). The membership degree of node v to community C is given by:

$$Bl(u,C) = \frac{1}{dist_{moy}(u,C) * \rho(u,C)} \tag{9}$$

where $dist_{moy}(u,C)$ is the average distance between u and C and $\rho(u,C)$ is the compactness of u to C.

From Eq. (9), we should notice that a node is "typical" to a community when "it is close to its members".

Property 5. *For all nodes of the graph G we have,* $0 \leqslant Bl(v,C) \leqslant 1$

Given all these definitions, now we are ready to outline the objective function of our model which will quantify the overall quality of partitioning.

3.3.3. Objective function

Our objective function called "Index of connectivity", is based on the idea that a community is simply a set of individuals with strong interactions between them and few interactions with the outside. This measure qualifies a partition based on both internal and external connections of its communities. But, before going any further we need the following definitions.

Definition 13 (*Compactness of a community*). It is the number of all the edges connecting the members of C. It is noted as follows $comp(C)$ and defined by:

$$comp(C) = \mid \{(v,v') \in E,\ v \in C \text{et} v' \in C\} \mid \tag{10}$$

Definition 14 (*Separability of a community*). It is the number of all the edges outside of community C. It is noted $sep(C)$ and defined by:

$$sep(C) = \mid \{(v,v') \in E,\ v \in C, v' \notin C\} \mid \tag{11}$$

The main objective is to compute a partitioning in which communities are compact and separable. However, the notion of separability does not imply total separability since overlaps are allowed. All these concepts are taken into consideration in our objective function "Index of connectivity" defined as follows.

Definition 15 (*Index of connectivity*). The index of connectivity of a community is based on the idea of maximizations of the internal links to a community. It is defined by the difference between the compactness of C and its separability, normalized by the sum of the compactness and the separability of the considered one. It is defined as follows:

$$IC(C) = \frac{comp(C) - sep(C)}{\sqrt{comp(C) + sep(C)}} \tag{12}$$

We notice that the principal of our objective function (the compute of "compactness" and "separability") was used in others models (Baumes et al., 2005; Lancichinetti et al., 2009) but not in the same way. Our objective function in Eq. (12) is maximal when the compactness of a community $comp(C)$ is high and its separability $sep(C)$ is low.

This simply means that the vertices included in this community have strong connections between them and are loosely connected with nodes outside this community. In an informal way, this means that we have a compact community separated from the rest by a sparse area.

Given our objective function, we will outline in the next section our algorithm DOCNet for its optimization and analyze its complexity.

4. Description of our method

With the correlative preliminaries above, we introduce an efficient algorithm, called **DOCNet** (**D**etection of **O**verlapping **C**ommunities in **N**etworks), to serve to the requirement for discovering overlapping communities of complex networks. This model is based on the principle of "agglomerative hierarchical clustering" since communities are built in an agglomerative manner. In fact, starting from a single node, we repeatedly expand its border nodes until it reaches an equilibrium state. Thus, DOCNet consists of two main components:

Algorithm 1. DOCNet(G)

Data: A graph $G = (V,E)$
Result: A set of overlapping communities $P = \{C_1, \ldots, C_n\}$.
begin
1 $P \leftarrow \emptyset$
for $i = 1 \mid V \mid$ **do**
2 $C_i \leftarrow \{n_i\}$.
$Imp \leftarrow \{n_i\}$.
end
3 Save importance of nodes on the vector *Imp*.
4 Sort nodes in *Imp* according to their importance in descending order.
5 **while** (there are vertices in *Imp*) **do**
6 Select the center c which is the first node in *Imp*.
7 Build the core of C: $C \leftarrow \{c\} \cup \{B(c)\}$.
8 **Extension(C, G)** .
9 $P \leftarrow P \cup C$.
10 Delete member of C from *Imp*.
end
11 Return (P).
end

Algorithm 2. Extension(C, G)

Data: A graph $G = (V,E)$, a community C
begin
1 Build border of C : $K_C \leftarrow \{n_i \mid n_i \in B(C)\}$.
2 **while** ($K_C \neq \emptyset$) **do**
3 Choose the candidate node n_c of K_C which has the highest membership degree to C.
4 **if** $IC(\{C\} \cup \{n_c\}) > IC(C)$ **then**
5 $C \leftarrow \{C\} \cup \{n_c\}$.
6 Update of K_C.
else
7 $K_C \leftarrow \emptyset$.
end
end
8 Return C.
end

1. Building the core of a community.
2. Extending its core.

DOCNet is outlined in Algorithm 1 this is explained in details in the following paragraph. We begin with an initial empty partition. As a starting point, we consider each vertex as a community. Then we compute the ("node importance") *NI* of all nodes of the graph. Then we sort vertices according to their importance and in descending order. These steps represent the initialization phase. The next step is the formation of the core of community. First, we select the most important node from vector *Imp*. In principal, this node is the "most influential" in the remaining non-partitional part of graph. Next, we build the "core of community" of *C*formed by its center and its border. Afterwards, we note by K_C the set of nodes situated on the border of C and which are candidates to its extension. Regarding the extension stage of C, we proceed as follows. We choose the candidate node n_c from K_C with the largest membership degree to the core community. We start by adding this node. If it increases our objective function we update all boundaries nodes and their membership degrees, and we check again the next vertex in K_C. Otherwise, we stop the expansion of this community, we remove community members from *Imp* and we move to the formation of the next community. This extension process is summarized in Algorithm 2. Summarizing, the extension of a community is stopped when the "most important node" of its boundary does not improve any further the objective function in Eq. (12). This implicitly means that if the most important boundary node does not improve the quality of the considered community; then no other boundary node can do so. This fundamental property of our extension process is guaranteed by the following theorem.

Theorem 2. *Let C be a community and K_C the set of its border nodes. Let $n_0 \in K_C$ be a node with the highest membership degree; i.e:* $arg(n_0) = max\{Bl(n_j, C) | n_j \in K_C\}$

Let $C' = C \cup \{n_0\}$. If the index of connectivity of C' is less than C, then any extension of C by any other node of K_C will no further its quality. Formally:

If $IC(C') < IC(C)$ **then**
$IC(C \cup \{n_j\}) < IC(C); \quad \forall n_j \in K_C$

Proof. We note, $d' = |B(n)|$ the degree of a node n in graph.

We have: $IC(C) = \frac{comp(C)-sep(C)}{\sqrt{comp(C)+sep(C)}}$.
Let $C' = C \cup \{n\}$

$$\begin{aligned}
\bullet comp(C') &= comp(C) + d_{in}(n, C)\\
\bullet sep(C') &= sep(C) - d_{in}(n, C) + d_{ext}(n, C)\\
&= sep(C) - d_{in}(n, C) + d' - d_{in}(n, C)\\
&= sep(C) + d' - 2d_{in}(n, C)
\end{aligned}$$

$$\begin{aligned}
Or\ IC(C') &= \frac{comp(C') - sep(C')}{\sqrt{comp(C') + sep(C')}}\\
&= \frac{comp(C) + d_{in}(n,C) - sep(C) - d' + 2d_{in}(n,C)}{\sqrt{comp(C) + d_{in}(n,C) + sep(C) + d' - 2d_{in}(n,C)}}\\
&= \frac{comp(C) - sep(C) - d' + 3d_{in}(n,C)}{\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)}}\\
&= \frac{-comp(C) - sep(C) - d' + d_{in}(n,C)) + 2comp(C) + 2d_{in}(n,C)}{\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)}}\\
&= -\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)}\\
&\quad + \frac{2comp(C) + 2d_{in}(n,C)}{\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)}}
\end{aligned}$$

$$Let \begin{cases} \bullet C_1 = C \cup \{n_1\}, & \text{where } n_1 \text{ is the node having the highest membership degree to } C.\\ \bullet C' = C \cup \{n\}, & \text{where } n \text{ is an arbitrary node of the boundary set } K_c \end{cases}$$

We want to prove that: $IC(C1) \leqslant IC(C) \rightarrow IC(C') \leqslant IC(C)$
Or we have:

$$\begin{cases} \bullet IC(C') = -\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)} + \dfrac{2comp(C) + 2d_{in}(n,C)}{\sqrt{comp(C) + sep(C) + d' - d_{in}(n,C)}}\\ \bullet IC(C_1) = -\sqrt{comp(C) + sep(C) + d_1 - d_{in}(n_1,C)} + \dfrac{2comp(C) + 2d_{in}(n_1,C)}{\sqrt{comp(C) + sep(C) + d_1 - d_{in}(n_1,C)}} \end{cases} \tag{13}$$

Observation

We notice that if the number of internal edges linking a node n to the community C increases then the sum of the shortest paths decreases (i.e., $d_{in}(n)$ and $\sum_{v \in C} dist(n, v)$ are inversely proportional) (see Fig. 1).

Or $Bl(n_1, C) \geqslant Bl(n, C)$.

We note d_{max} the maximal degree of node in the graph. So we have $d \leqslant d_{max}$, $\forall n \in V$ and specially, $d_1 \leqslant d_{max}$ and $d' \leqslant d_{max}$. We assume here that we are in the extreme case then; $d' = d_1 = d_{max} = d$.

So it is enough to check that $d_{in}(n_1, C) \geqslant d_{in}(n, C)$

$$\Rightarrow \frac{|C|\, d_{in}(n_1,C)}{d\sum_{v\in C} dist(n_1, v)} \geqslant \frac{|C|\, d_{in}(n,C)}{d\sum_{v\in C} dist(n, v)} \Longleftrightarrow \frac{d_{in}(n_1,C)}{\sum_{v\in C} dist(n_1, v)} \geqslant \frac{d_{in}(n,C)}{\sum_{v\in C} dist(n, v)}$$

We suppose that

$$d_{in}(n_1, C) \leqslant d_{in}(n, C)$$

$$\Rightarrow \sum_{v\in C} dist(n_1, v) \geqslant \sum_{v\in C} dist(n, v) \Rightarrow \frac{1}{\sum_{v\in C} dist(n_1, v)} \geqslant \frac{1}{\sum_{v\in C} dist(n, v)}$$

$$\Rightarrow \frac{d_{in}(n_1)}{\sum_{v\in C} dist(n_1, v)} < \frac{d_{in}(n)}{\sum_{v\in C} dist(n, v)}$$

$\Rightarrow$ this is contradictory.
Therefore

$$d_{in}(n_1, C) \geqslant d_{in}(n, C) \tag{14}$$

And subsequently $d - d_{in}(n_1, C) \leq d - d_{in}(n, C)$
Hence:

$$\sqrt{comp(C) + sep(C) + (d - d_{in}(n_1, C))} \leqslant \sqrt{comp(C) + sep(C) + (d - d_{in}(n, C))} \tag{15}$$

We suppose that $IC(C') \geq IC(C)$

$$\Rightarrow -\sqrt{comp(C) + sep(C) + d - d_{in}(n,C)} + \frac{2comp(C) + 2d_{in}(n,C)}{\sqrt{comp(C) + sep(C) + d - d_{in}(n,C)}} \geqslant IC(C) \tag{16}$$

According to (14) we have:

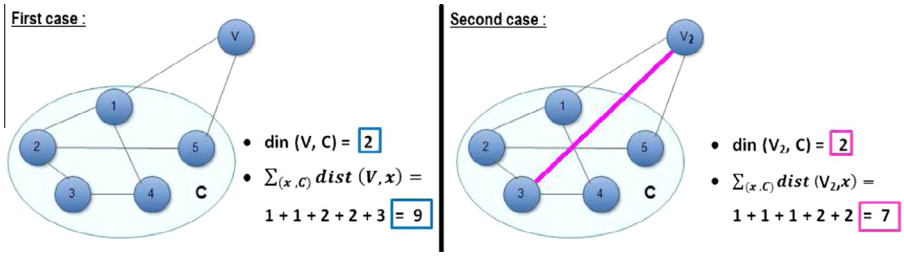

Fig. 1. Relationship between the internal degree of node to a community and the shortest path to all vertices.

$$-\sqrt{comp(C)+sep(C)+d-d_{in}(n,C)} + \frac{2comp(C)+2d_{in}(n_1,C)}{\sqrt{comp(C)+sep(C)+d-d_{in}(n,C)}} \geqslant IC(C) \quad (17)$$

In accordance with (15) we have:

$$-\sqrt{comp(C)+sep(C)+d-d_{in}(n_1,C)} + \frac{2comp(C)+2d_{in}(n_1,C)}{\sqrt{comp(C)+sep(C)+d-d_{in}(n_1,C)}} \geqslant IC(C) \quad (18)$$

$\Rightarrow IC(C_1) \geqslant IC(C)$

$\Rightarrow$ this is contradictory.

Therefore

$IC(C') \leqslant IC(C)$

Thus

$IC(C1) \leqslant IC(C) \Rightarrow IC(C') \leqslant IC(C)$ □

Concluding, Theorem 2 gives us a fundamental criteria for stopping the expansion of a community. Our algorithm DOCNet stops when there are no "free nodes" (i.e. nodes that are not yet assigned to any communities) in the graph.

An important element in the detection of overlapped communities in large-scale social networks are both time and space complexity of approach. For this, we will evaluate theoretically the performance of our algorithm by computing its temporal and spatial complexity. The following theorem is about the time complexity of *DOCNet*.

Theorem 3. *The time complexity of our algorithm DOCNet is $O(n^2)$ where n is the number of vertices in the social network*

Proof. To calculate the time complexity of our algorithm, we will handle step by step. We begin by the step of construction of the core of community, which is as follows: first, the initial partition is done in $O(n)$. The time complexity of the build of vector *Imp*, in which we calculate the shortest path between all vertices of graphs, is $O(m+nlog(n))$ and the degree of all nodes on $O(n^2)$. Then we sort the vector *Imp*by quicksort with $O(nlog(n))$. After, we select the center of the core and we form community composed by this node and its direct neighbors. This step is made in $O(d_{max})$, with d_{max} is the maximum degree of a node in *G*. So the time complexity of the first step is:

$$T_{Step1} = O(max\{n^2, nlog(n), d_{max}\}) = O(n^2)$$

Then we move to the second stage which is the extension of the core of community, it is defined as follows: first the construction of the initial border with a complexity of $O(d^2_{max})$, The index of connectivity is made in $O(d_{max}(1+d_{max}))$. Choosing node with the maximum belonging degree from the border of *C* is done *T* time (*T* is the border size and it is *n* in the worst case) so its complexity is $O(Tn)$. All steps mentioned above are repeated in the worst case *c* times with *c* is the number of communities.

$$T_{Step2} = O\left(max\left\{c(d^2_{max}), cd_{max}(1+d_{max}), cn^2\right\}\right) = O(cn^2)$$

Finally we note that in the general case the number of communities *c* is negligible compared to *n*. Therefore, the time complexity of the entire algorithm can be estimated to be:

$$T_{Temporal}(DOCNet) = O(n^2) \quad \square$$

Theorem 4. *The space complexity of our algorithm DOCNet is $O(m)$ where m is the number of edges in the social network.*

Proof. In our algorithm, we use the vector *Imp* that contains all the vertices of graph sorted according to their importance. The size of this vector is *n* with *n* is the number of vertices. We also build a graph with *m* edges. Hence the total space complexity of our algorithm is:

$$T_{Space}(DOCNet) = O(max\{m,n\}) \quad = O(m). \quad \square$$

In summary, we have presented an objective function to qualify the overall quality of a computed partitioning. Moreover, we have presented DOCNet, an algorithm for its optimization. The basic idea of DOCNet is to expand a community from "most influantial" nodes until no further improvements in the objective function can be made. the complexity of DOCNet turned to be quadratic in time and linear in space. These could be considered as satisfactory measures. The next section will evaluate our model on large-scale and real-world social networks.

5. Experimental results

The main purpose of this section is to analyze the behavior of DOCNet experimentally.[1] For this, we conducted extensive simulation on both synthetic and three real-world networks. We compared DOCNet with three well-known algorithms: (1) **CFinder** (CPM) which implements the clique percolation (Palla, 2011); (2) **COPRA** which is based on label propagation (Gregory, 2010); (3) **GCE** greedy approach (GCE, 2013); and (4) **EAGLE** modularity-based approach (Eagle Community Detection Algorithm, 2012). Simulation results on synthetic and real-world networks are presented subsequently. Before reporting the experimental results, we need to indicate the performance indices that we will adopt.

[1] Our proposed algorithm is implemented in Java and ran on a PC (Intel Core i5 Quad CPU 3.2 GHz with 15.0 GB of memory).

5.1. Evaluation criterion

5.1.1. Normalized mutual information (NMI)

We adopted the extended normalized mutual information (NMI) which takes into account the overlap between communities. It is proposed by Lancichinetti et al. (2009) and yields values between 0 and 1, with 1 corresponds to a perfect case. NMI between two partition C' and C'' (respectively the expected and the real community structure) is given by:

$$NMI(X \mid Y) = 1 - \frac{[H(X|Y) + H(Y|X)]}{2} \tag{19}$$

The conditional entropy of a cluster X_k given Y_l is defined as: $H(X_K \mid Y_l) = H(X_k, Y_l) - H(Y_l)$. 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 $H(X_k \mid Y) = min_{l \in \{1,2,\ldots,|C''|\}} H(X_k \mid Y_l)$.

The normalized conditional entropy of a partition X with respect to Y is: $H(X|Y) = \frac{1}{|C'|}\sum_k \frac{H(X_k|Y)}{H(X_k)}$.

5.1.2. F-score

To provide more precise analysis, we consider the identification of overlapping nodes given by f-score:

$$F = \frac{2 * precision * recall}{precision + recall} \tag{20}$$

where **recall** is the number of correctly detected overlapping nodes divided by the true number of overlapping nodes and **precision** is defined as the number of correctly detected overlapping nodes divided by the total number of detected overlapping nodes (Fortunato, 2010).

5.1.3. The modularity of overlap

This is an extension of the classical modularity (Shen et al., 2008). It takes into account the number of communities to which each vertex belongs and the degree of membership in each community.

$$Q_{ov} = \frac{1}{2m}\sum_c \sum_{i,j \in c}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\frac{1}{O_i O_j} \tag{21}$$

where O_i is the number of communities to which the node i belongs, m is the number of edge of the graph, k_i is the degree of a node i and A_{ij} is the element of adjacency matrix. A good partitioning maximizes modularity.

5.2. Artificials networks

We adopted the LFR (Benchmark Graphs, 2013) benchmark for synthetic networks for which the exact partition of the network is known. In our experiments, we used five sizes of networks $N = 1000$, $N = 2000$, $N = 5000$, $N = 8000$, $N = 10{,}000$ and $N = 50{,}000$. The rest of the parameters are as follows: The average degree is kept at $k = 10$; the mixing parameter μ is 0.2; the maximum degree is 50; the community size varies between 20 and 100; O_n (the number of overlapping nodes) varies from 10% to 80%; O_m (the number of communities to which each overlapping node belongs) varies from 2 to 8. By increasing the value of O_n or O_m, we create harder detection tasks.

5.2.1. DOCNet performance with the variation of network complexity

We generated a number of graphs while varying μ which models the fraction of edges outside the community of each node relative to the total number of edges. In fact, as this parameter increases, the boundaries between communities become less clear. The results are illustrated in Fig. 2 and we note from these curves that the increase in the ratio μ has an influence on the quality of partition detected by all models studied. In fact, as μ increases, the NMI decreases. This means that the network becomes more complex, the calculated partition is away from the exact one. The CPM and EAGLE models still farthest from the exact solution. COPRA has a good quality for simple graphs and tends sharply to zero solution. As the network complexity increases, our model retains a uniform behavior even when it is a complex case in which each node have many links within the community and outside it. GCE gives the best result. As against, our model is better than CPM, EAGLE and COPRA especially for complex networks (i.e., when μ increases).

5.2.2. DOCNet performance with graphs of thousand nodes

The first LFR benchmark contains 1000 nodes. The community size ranges from 20 to 100. The mixing parameter μ is 0.2. Simulation results are reported in Tables 1–4 and Fig. 3. According to the results of Tables 1–3 and Fig. 3 we note that the NMI, the recall, precision and F-score of COPRA tend sharply to zero values. GCE is too close to the exact partition but it has a limited recall. CPM represent good precision but a faraway partition from the real one. Similar to CPM, EAGLE achieves a high precision and a low NMI when varying On. Our model maintains a nearly constant quality partition influenced by the increase of O_n. It has the best recall and F-score. The experimental results are shown in Table 4. GCE has the best quality of partition. Against by, CPM is the farthest from the real structure of the graphs. Our model has an aspect too close to the exact one compared with COPRA and GCE.

5.2.3. DOCNet performance with graphs of two thousand nodes

In this second set of runs, we consider LFR graph containing 2000 nodes, the mixing parameter is 0.2 and communities' size vary between 20 and 100. Simulation results are summarized in Tables 5–8 and Fig. 4. Similar results were obtained. Based on the results in Table 5, we observe that GCE has the best partition by varying the number of overlapped nodes (O_n) because it uses maximal cliques as seed nodes, which are easy to find in such dense graph. The second is DOCNet. CPM, EAGLE and COPRA are more distant and tend rapidly to zero values. According to the results of Tables 6, 7 and Fig. 4 we observe that DOCNet has the best recall. We also find that the F-score is the most significant. The data reported in Table 8 show that when varying O_m from 2 to 8; DOCNet retains a good quality partition, which varies between 0.80 and 0.43 and it is close to GCE which is the best.

5.2.4. DOCNet performance with graphs of five thousands nodes

We consider graphs of 5000 nodes and when varying O_m and O_n we obtain the following results: We denote from Tables 9–11 that DOCNet is influenced by the increase of overlap, like the other

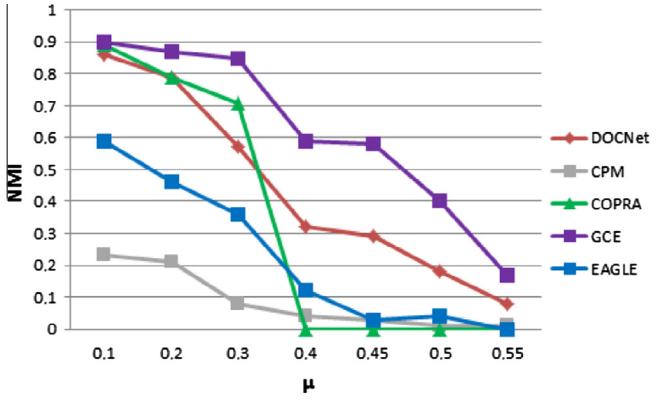

Fig. 2. Variation of NMI in function of μ.

models. In fact, increasing this rate causes a decrease in the quality of partition. But we notice that our model has a good quality of NMI compared to other.

From Fig. 5, we remark that since the rate of overlap is more than 45% DOCNet manages to have the best F-score than the other methods. In fact, it has an evolutionary aspect which increases with O_n, contrary to GCE. Although our model has shown its effectiveness (in terms of NMI) by increasing the rate of overlap O_n, it fails to be the first by comparing it with the best known methods and the most recent. In fact, its quality is the best in taking $O_m = 2$ (see Table 12).

5.2.5. DOCNet performance with graphs of eight thousands nodes

For the LFR-8000 graph and from Table 13 we find that with the change in overlap rate our model lowers the quality of partition which remained always better than CPM, EAGLE and COPRA. On the other hand, DOCNet and as shown in Table 14 is able to have better recall than the other algorithms. The Fig. 6 shows the F-score as a function of the rate of overlapping nodes. Since O_n is more than 40%, DOCNet achieves the largest F-score in networks with different size. It has also an evolutionary aspects. GCE, EAGLE and CPM have average performance. On the contrary of COPRA which is the worst. We notice that our model has a positive correlation with O_n (see Table 15). While other algorithms demonstrate a negative correlation. This is due to the high recall of DOCNet.

According to the results of Table 16 we notice that since the rate of overlap is more than 20%, DOCNet manages to have the best NMI than the other methods. In fact, it has an evolutionary aspect which increases with O_m, unlike of GCE, COPRA, EAGLE and CPM. This is even the case for $N = 8000$ and large O_n. So, we observe that our model provides a good partition for large scale graph.

5.2.6. DOCNet performance with graphs ten thousands nodes

In this type of graph with large size and from Table 17, we find that COPRA found the best solution by varying the overlap rate from 10% to 30% and it tends sharply to zero value. The CGE model found a good partition throughout the change of O_n. In regarding our model, it has an average aspect. In fact, it is better than CPM for performance concerning the NMI. By observing other measures of performance, we note that our model has the highest recall, its precision is average and F-score increases by incrementing the overlap rate. EAGLE suffers from under-detection (where only very few overlapping nodes are identified) which results in a low recall score. This is clear in Tables 18, 19 and Fig. 7. In this case, from Table 20, we find that our algorithm has a quality of partition that improves by increasing the number of communities to which node belongs. COPRA and GCE are the best. From Fig. 8 DOCNet has the best F-score by varying the O_m.

5.2.7. DOCNet performance with graphs fifty thousands nodes

As a final test for artificial networks, we generated graphs with large size. Simulation results are summarized in Table 21. We notice from these results that our model was in average able to detect

Table 1
NMI for networks with $N = 1000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.61	0.90	0.33	0.37	0.77
20	0.58	0.85	0.30	0.43	0.68
30	0.55	0.80	0.08	0.32	0.58
40	0.00	0.59	0.07	0.08	0.41
50	0.00	0.56	0.06	0.13	0.39
60	0.00	0.41	0.06	0.05	0.26
70	0.00	0.34	0.06	0.02	0.23
80	0.00	0.28	0.02	0.02	0.17

Table 2
Recall for networks with $N = 1000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.84	0.37	0.18	0.45	0.73
20	0.78	0.30	0.21	0.39	0.68
30	0.76	0.45	0.07	0.43	0.72
40	0.00	0.23	0.17	0.35	0.71
50	0.00	0.57	0.10	0.36	0.70
60	0.00	0.39	0.14	0.30	0.78
70	0.00	0.44	0.13	0.32	0.76
80	0.00	0.54	0.17	0.33	0.77

Table 3
Precision for networks with $N = 1000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.37	0.94	0.38	0.43	0.30
20	0.55	0.85	0.53	0.69	0.36
30	0.71	0.83	0.43	0.68	0.45
40	0.00	0.68	0.61	0.80	0.49
50	0.00	0.70	0.67	0.82	0.61
60	0.00	0.67	0.74	0.84	0.62
70	0.00	0.78	0.79	0.82	0.73
80	0.00	0.83	0.84	0.93	0.80

Table 4
NMI for networks with $N = 1000,\ K = 10,\ \mu = 0.2,\ O_n = 10\%$.

O_m	COPRA	GCE	CPM	EAGLE	DOCNet
2	0.61	0.90	0.33	0.44	0.74
3	0.56	0.80	0.00	0.45	0.62
4	0.65	0.73	0.30	0.39	0.54
5	0.63	0.73	0.38	0.28	0.50
6	0.00	0.69	0.27	0.23	0.44
7	0.00	0.65	0.34	0.21	0.43
8	0.00	0.62	0.34	0.23	0.42

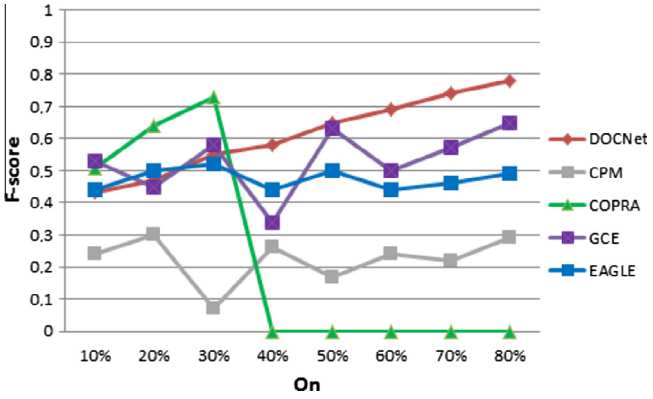

Fig. 3. F-score for networks with $N = 1000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

Table 5
NMI for networks with $N = 2000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.85	0.89	0.49	0.40	0.78
20	0.72	0.82	0.40	0.35	0.74
30	0.58	0.81	0.27	0.15	0.59
40	0.44	0.77	0.25	0.10	0.51
50	0.00	0.68	0.14	0.12	0.45
60	0.00	0.58	0.09	0.03	0.32
70	0.00	0.32	0.06	0.02	0.19
80	0.00	0.26	0.04	0.10	0.07

Table 6
Recall for networks with $N = 2000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	AGLE	DOCNet
10	0.93	0.38	0.40	0.36	0.65
20	0.81	0.36	0.29	0.40	0.73
30	0.68	0.88	0.19	0.35	0.71
40	0.57	0.42	0.23	0.36	0.73
50	0.00	0.50	0.25	0.34	0.75
60	0.00	0.54	0.17	0.33	0.74
70	0.00	0.45	0.17	0.21	0.75
80	0.00	0.48	0.21	0.26	0.76

Table 7
Precision for networks with $N = 2000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.39	0.68	0.51	0.67	0.31
20	0.54	0.77	0.72	0.81	0.41
30	0.67	0.88	0.66	0.74	0.41
40	0.71	0.86	0.77	0.84	0.48
50	0.00	0.72	0.77	0.86	0.56
60	0.00	0.76	0.80	0.84	0.64
70	0.00	0.75	0.80	0.86	0.70
80	0.00	0.83	0.89	0.33	0.82

Table 8
NMI for networks with $N = 2000$, $K = 10$, $\mu = 0.2$, $O_N = 10\%$.

O_m	COPRA	GCE	CPM	EAGLE	DOCNet
2	0.85	0.90	0.42	0.42	0.80
3	0.64	0.79	0.52	0.40	0.57
4	0.55	0.74	0.43	0.36	0.51
5	0.53	0.72	0.41	0.35	0.47
6	0.46	0.69	0.45	0.44	0.46
7	0.41	0.65	0.36	0.17	0.41
8	0.32	0.60	0.39	0.30	0.43

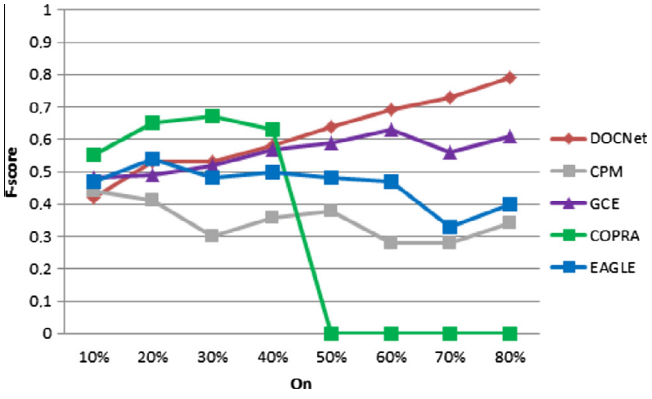

Fig. 4. F-score for networks with $N = 2000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

Table 9
NMI for networks with $N = 5000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	AGLE	DOCNet
10	0.79	0.79	0.57	0.39	0.82
20	0.65	0.87	0.54	0.18	0.69
30	0.53	0.82	0.42	0.17	0.64
40	0.42	0.76	0.24	0.06	0.51
50	0.36	0.67	0.17	0.17	0.41
60	0.25	0.52	0.14	0.15	0.33
70	0.00	0.40	0.12	0.15	0.27
80	0.00	0.32	0.12	0.14	0.19

Table 10
Recall networks with $N = 5000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.94	0.33	0.45	0.37	0.68
20	0.80	0.36	0.48	0.38	0.72
30	0.66	0.40	0.44	0.39	0.69
40	0.54	0.38	0.38	0.36	0.70
50	0.47	0.47	0.36	0.31	0.75
60	0.42	0.51	0.37	0.28	0.75
70	0.00	0.47	0.32	0.26	0.74
80	0.00	0.47	0.30	0.23	0.75

Table 11
Precision for networks with $N = 5000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.39	0.93	0.56	0.78	0.37
20	0.59	0.90	0.79	0.85	0.39
30	0.66	0.87	0.84	0.92	0.46
40	0.69	0.84	0.83	0.89	0.48
50	0.74	0.76	0.88	0.89	0.57
60	0.79	0.73	0.88	0.90	0.63
70	0.00	0.78	0.89	0.90	0.72
80	0.00	0.84	0.91	0.91	0.81

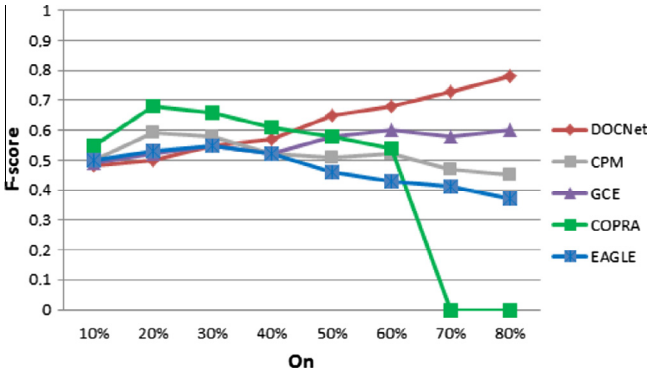

Fig. 5. F-score for networks with $N = 5000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

Table 12
NMI for networks with $N = 5000$, $K = 10$, $\mu = 0.2$, $O_N = 10\%$.

O_m	COPRA	GCE	CPM	EAGLE	DOCNet
2	0.79	0.92	0.57	0.39	0.81
3	0.67	0.79	0.51	0.30	0.58
4	0.50	0.77	0.53	0.35	0.53
5	0.52	0.71	0.44	0.36	0.48
6	0.52	0.67	0.44	0.42	0.45
7	0.45	0.63	0.43	0.27	0.42
8	0.35	0.60	0.43	0.25	0.40

Table 13
NMI for networks with $N = 8000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.74	0.91	0.67	0.31	0.84
20	0.66	0.86	0.55	0.24	0.70
30	0.57	0.57	0.49	0.13	0.62
40	0.44	0.44	0.38	0.07	0.51
50	0.35	0.35	0.28	0.10	0.41
60	0.28	0.28	0.26	0.11	0.36
70	0.20	0.20	0.21	0.09	0.25
80	0.00	0.18	0.00	0.07	0.19

Table 14
Recall for networks with $N = 8000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.92	0.35	0.50	0.37	0.11
20	0.81	0.34	0.48	0.38	0.68
30	0.65	0.35	0.49	0.41	0.73
40	0.58	0.33	0.43	0.36	0.72
50	0.47	0.46	0.40	0.32	0.78
60	0.37	0.46	0.55	0.30	0.73
70	0.34	0.47	0.37	0.27	0.74
80	0.00	0.48	0.00	0.20	0.74

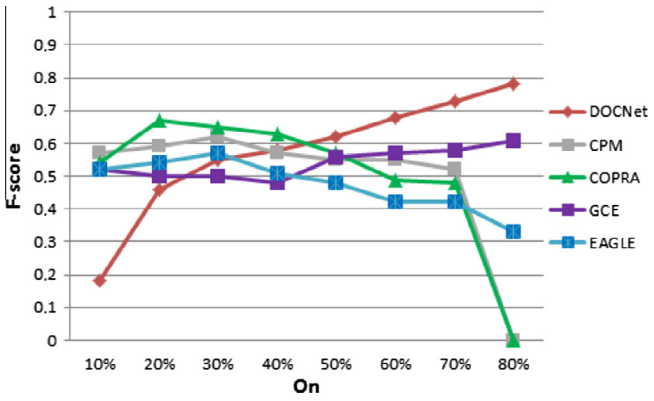

Fig. 6. F-score for networks with $N = 8000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

the exact structure for each graph despite its size. We remark also that COPRA and GCE compute a good partition which is not so far from the exact one. However, the F-score of DOCNet was always the highest. This is caused by its excessively high recall. Similar to small size network, COPRA and GCE maintain great precision score. Regarding CPM and EAGLE, we were unable to test them on these complex graphs due to their computation inefficiency.

5.2.8. DOCNet performance with the variation of network density

We evaluated in this section the effect of the variation of the number of edges (we increased the density of the graph), on networks with $N = 8000$ and $\mu = 0.2$. Quantities reported by DOCNet in Fig. 9 are closer to the ground truth and are slightly steady during the variation of k. From $k = 40$ we are unable to run neither CPM nor EAGLE due to their time-complexity. GCE's NMI outperforms other algorithms when the density of the network rises. COPRA has an increasing aspect and a positive correlation with the variation of the number of edges. Plots of Fig. 10 shows F-score for networks with the increase of network size (in term of number of edges) from 40,948 ($k = 10$) to 319,714 ($k = 80$). GCE as well as DOCNet achieves nearly the largest F-score in networks with high level of density, as defined by k. Interestingly, COPRA, GCE and DOCNet have a positive correlation with k (they made a harmonic balance between precision and recall) while other algorithms typically demonstrate a negative correlation.

Table 15
Precision for networks with $N = 8000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.38	0.96	0.68	0.87	0.40
20	0.57	0.93	0.78	0.90	0.35
30	0.64	0.89	0.84	0.91	0.44
40	0.70	0.82	0.84	0.91	0.48
50	0.73	0.74	0.87	0.91	0.55
60	0.76	0.75	0.88	0.92	0.63
70	0.82	0.77	0.90	0.92	0.72
80	0.00	0.84	0.00	0.94	0.82

Table 16
NMI for networks with $N = 8000,\ K = 10,\ \mu = 0.2,\ O_n = 10\%$.

O_m	COPRA	GCE	CPM	EAGLE	DOCNet
2	0.74	0.91	0.70	0.31	0.84
3	0.62	0.80	0.60	0.34	0.58
4	0.61	0.77	0.56	0.33	0.53
5	0.54	0.71	0.53	0.36	0.49
6	0.46	0.68	0.50	0.26	0.44
7	0.40	0.63	0.51	0.30	0.42
8	0.39	0.58	0.46	0.38	0.37

Table 17
NMI for networks with $N = 10,000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.74	0.42	0.18	0.58	0.28
20	0.71	0.21	0.12	0.52	0.20
30	0.62	0.24	0.10	0.40	0.18
40	0.00	0.32	0.08	0.29	0.16
50	0.00	0.19	0.03	0.10	0.08
60	0.00	0.18	0.01	0.03	0.08

As a conclusion to these simulations of artificial networks, we can state that our model performs well for distinct types of graphs and was able to compute the practically exact structure of each network regardless of its nature and complexity. The F-score of all tested methods typically decays moderately as overlapping diversity O_m increases. The high precision of EAGLE (also CPM and GCE for $O_m = 2$) shows that clique-like assumption of communities may help to identify overlapping nodes in low overlapping density case. DOCNet has a great capacity to detect the boundary node (high F-score). However, GCE achieves the highest NMI than other models. Subsequently, we will continue our experimental analysis but considering real networks.

5.3. Reals networks

After having considered artificial networks, we will consider actually well-known real social networks used in the literature as a test for several models. Unfortunately, not all of these models do have an exact known reference structure. Some of used measures to evaluate the performance of overlapping community detection in real-world networks are the number of exacted detected communities (denoted as M), the number of detected overlapping nodes (O_n^d) and the average number of detected memberships (O_m^d). We tested algorithms for the detection of community overlap on eight reals social networks. The Table 22 provides some real benchmark networks. We removed CPM and EAGLE from the test for (PGP, ca-CondMat, ca-CondMat) due to either their memory or computation inefficiency in large networks (Xie et al., 2013).

5.3.1. Karate Club

The social network of Karate Club members studied by the sociologist has become a famous benchmark for all community

Table 18
Recall for networks with $N = 10,000,\ K = 10,\ \mu = 0.2,\ O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.49	0.05	0.11	0.05	0.67
20	0.57	0.07	0.10	0.04	0.65
30	0.41	0.05	0.11	0.04	0.67
40	0.00	0.06	0.13	0.06	0.67
50	0.00	0.07	0.12	0.02	0.69
60	0.00	0.08	0.12	0.04	0.68

Table 19
Precision for networks with $N = 10,000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

O_n (%)	COPRA	GCE	CPM	EAGLE	DOCNet
10	0.49	0.08	0.09	0.30	0.10
20	0.45	0.45	0.15	0.37	0.20
30	0.49	0.49	0.26	0.50	0.29
40	0.00	0.00	0.38	0.62	0.39
50	0.00	0.00	0.46	0.08	0.49
60	0.00	0.00	0.59	0.77	0.58

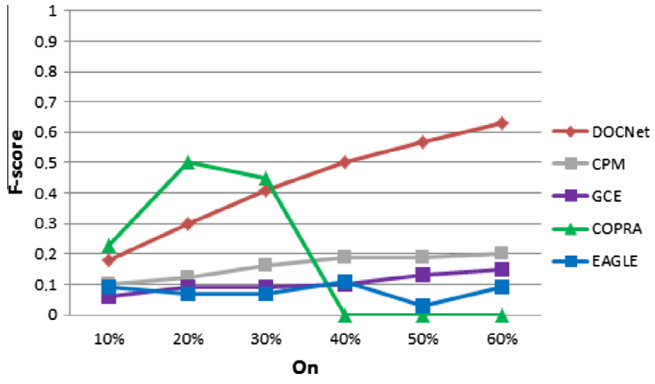

Fig. 7. F-score for networks with $N = 10,000$, $K = 10$, $\mu = 0.2$, $O_m = 2$.

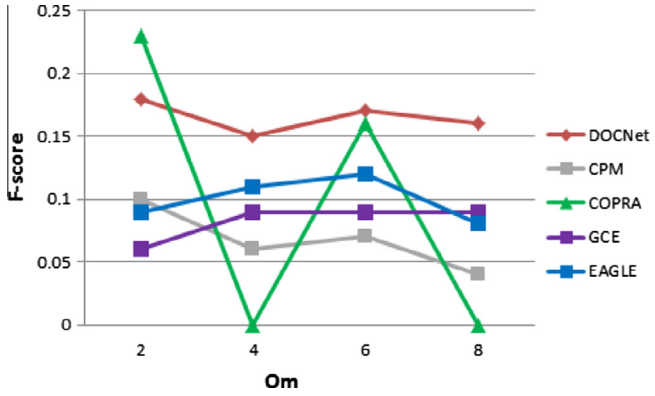

Fig. 8. F-score for networks with $N = 10,000$, $K = 10$, $\mu = 0.2$, $O_n = 10\%$.

Table 20
NMI for networks with $N = 10,000$, $K = 10$, $\mu = 0.2$, $O_n = 10\%$.

O_m	COPRA	GCE	CPM	EAGLE	DOCNet
2	0.74	0.42	0.18	0.59	0.28
3	0.70	0.32	0.18	0.58	0.21
4	0.00	0.14	0.10	0.57	0.14
5	0.00	0.34	0.14	0.59	0.22
6	0.00	0.33	0.14	0.45	0.27
7	0.00	0.43	0.17	0.32	0.28
8	0.00	0.45	0.13	0.40	0.25

Table 21
Performances of networks with $N = 50,000$, $K = 10$, $\mu = 0.2$, $O_n = 10\%$, $O_m = 2$, $|E| = 250,000$.

	COPRA	GCE	CPM	EAGLE	DOCNet
NMI	0.92	0.80	–	–	0.76
Precision	0.50	0.30	–	–	0.90
Recall	0.98	0.91	–	–	0.55
F-score	0.67	0.45	–	–	0.68

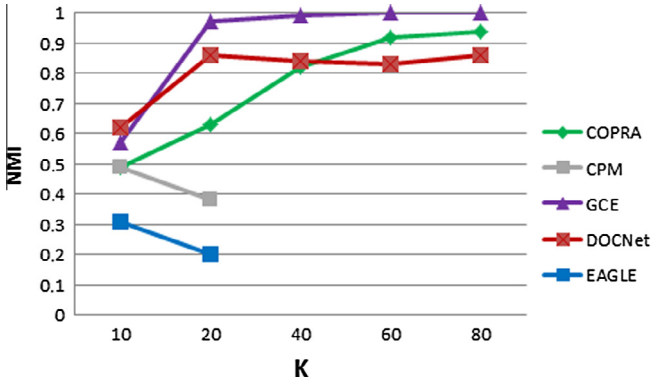

Fig. 9. Variation of NMI in function of k.

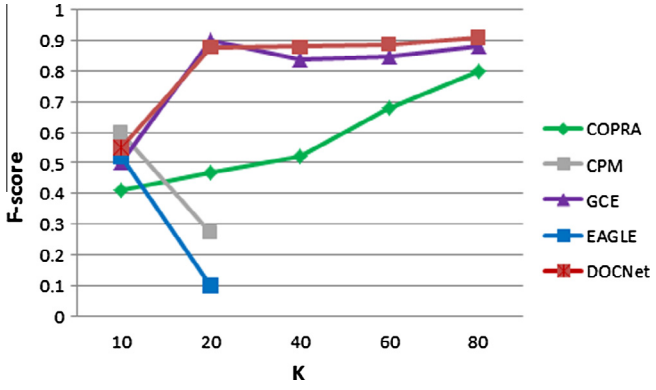

Fig. 10. Variation of F-score in function of k.

detection methods (Zachary, 1977). The network consists of 34 nodes, each node represents a member of the club, separated into two distinct groups centered around instructors or administrators. A partition into three communities is also justified. The experimental results are listed in Table 23.

5.3.2. Dolphin network

The Dolphins social network is taken from (Lusseau et al., 2003). It describes the associations between 62 dolphins living in Doubtful Sound, New Zealand. Ties between dolphin represent the statistically significant frequent association between them. The experimental results are listed in Table 24.

5.3.3. Les Miserables network

The network of "Les Miserables" takes into account interactions between the main characters in the novel written by Victor Hugo "Miserables" (Knuth, 1993). In this network, nodes represent characters and a link between two nodes is the simultaneous occurrence of two or more characters in a scene. Five communities are detected by our method which is similar to the exact one. The modularity of overlap is 0.29. More detailed results are reported in Table 25.

5.3.4. Books network

The Network Krebs books on U.S. policy is introduced by Newman. In this network (Krebs, 2004), nodes represent 105 recent books on American politics bought from Amazon.com. Links join pairs of books that are often purchased by the same buyer. Our score is perfectly the same as the actual partition that is three communities: "liberal", "neutral" or "conservative". Table 26 is a set of evaluation criteria for our model as well as 4 others.

Table 22
The characteristics of real graphs.

Networks	$\|V\|$	$\|E\|$	Description
Karate (Network data, 2013a)	34	78	The Zachary Karate Club
Dolphin (Network data, 2013a)	62	159	The network of dolphins
Miserables (Network data, 2013a)	77	254	The network of miserables
Books (Network data, 2013a)	105	441	The network of American Politics Books
email (Network data, 2013b)	1133	5451	The E-mail network URV
GR-QC (SNAP, 2009)	5242	14496	General Relativity and Quantum Cosmology
PGP (Network data, 2013b)	10680	24316	Pretty-Good-Privacy
Ca-CondMat (Network data, 2013a)	40421	175692	Collaboration network of Arxiv Condensed Matter

5.3.5. Email network

Email network describes the e-mail interchanges between members of the Univeristy Rovira Virgili (Tarragona). In fact, nodes represent members and an edge exists if two members interchange e-mail. The performance of various models is listed in Table 27. Algorithms may not perform equally well on different types of network structures. For example GCE, it is sensitive to specific structures. As shown in Table 27 CPM has the smallest fraction of overlapping nodes. Regarding COPRA, it detected two large communities and achieves the largest Q_{ov}.

5.3.6. GR-QC network

General Relativity and Quantum Cosmology collaboration network is from the e-print arXiv and covers scientific collaborations between authors papers submitted to General Relativity and Quantum Cosmology category (SNAP, 2009). If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. The experimental results listed in Table 28 shown that our proposal achieves the highest overlapped nodes. EAGLE and CPM perform worse than either GCE or COPRA.

5.3.7. PGP network

The final example is the network of PGP (Network data, 2013b), it is composed from 10680 nodes. This network is a giant list of users algorithm Pretty-Good Privacy (PGP software digit surely and tearing Cryptographic rately created by the American Phil Zimmermann in 1991). It guarantees the confidentiality and authentication of data communication for a secure exchange of information. The partition of this network is not known in advance but from the results in Table 29 CPM fails even to partition the graph because of its large size and complexity.

5.3.8. Ca-CondMat network

Condense Matter Physics is a collaborative network (SNAP, 2009). It is from the e-print arXiv and covers scientific collaborations between authors papers submitted to Condense Matter category. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. The data represents essentially the complete history of COND-MAT section (SNAP, 2009). As shown in Table 30, GCE achieves the highest Q_{ov}, however DOCNet outperforms others models significantly in term of O_m^d and O_n^d. CPM and EAGLE fail to partition the graph because of its large size and complexity.

Table 23
The quality measures partitioning of Karate Club obtained for the various models.

	CPM ($k=3$)	COPRA ($v=3$)	GCE	EAGLE	DOCNet
Q_{ov}	0.32	0.04	0.26	0.31	0.24
M	3.00	6.00	2.00	4.00	3.00
O_m^d	2.00	2.00	2.00	2.00	2.00
O_n^d	1.00	2.50	3.00	1.00	5.00

Table 24
The quality measures partitioning of Dophins obtained for the various models.

	CPM ($k=3$)	COPRA ($v=3$)	GCE	EAGLE	DOCNet
Q_{ov}	0.29	0.32	0.33	0.32	0.41
M	4.00	3.00	4.00	4.00	3.00
O_m^d	2.00	2.00	2.00	2.00	2.00
O_n^d	2.00	1.75	2.00	1.50	1.66

Table 25
The quality measures partitioning of miserable obtained for the various models.

	CPM ($k=4$)	COPRA ($v=2$)	GCE	EAGLE	DOCNet
Q_{ov}	0.32	0.37	0.34	0.20	0.29
M	4.00	4.00	5.00	16.00	5.00
O_m^d	2.33	2.00	2.00	2.22	2.52
O_n^d	1.25	2.00	1.25	2.60	6.86

Table 26
The quality measures partitioning of books obtained for the various models.

	CPM ($k=3$)	COPRA ($v=2$)	GCE	EAGLE	DOCNet
Q_{ov}	0.39	0.45	0.40	0.30	0.45
M	4.00	2.00	5.00	7.00	3.00
O_m^d	2.00	2.00	0.00	2.00	2.00
O_n^d	2.25	2.00	0.00	3.00	2.00

Table 27
The quality measures partitioning of email obtained for the various models.

	CPM ($k=3$)	COPRA ($v=3$)	GCE	EAGLE	DOCNet
Q_{ov}	0.37	0.49	0.18	0.17	0.48
M	41.00	2.00	35.00	44.00	30.00
O_m^d	2.12	2.00	2.21	2.59	2.66
O_n^d	1.92	8.00	2.30	2.39	10.00

To recapitulate, we can say that DOCNet performs well with respect to the adopted criteria, on real-world small and large-scale social networks. In fact, it was always able to compute the partition regardless of the size of the network or the size of the communities. As it is shown in Tables 23–30, COPRA achieves the highest Q_{ov} in nearly all the test networks. DOCNet has a moderate one. On average, EAGLE and CPM perform worse than either COPRA or GCE. It is also, interesting to note that all models confirm that the diversity of overlapping nodes in the tested real social networks is small (O_m^d is close to 2). Although the number of overlapping nodes differs from algorithm to others. DOCNet seems to be appropriated to the concept of overlap and returns higher number of overlapping nodes than others algorithms.

Table 28
The quality measures partitioning of GR-QC obtained for the various models.

	CPM ($k = 4$)	COPRA ($v = 3$)	GCE	EAGLE	DOCNet
Q_{ov}	0.23	0.24	0.26	0.10	0.23
M	835.00	405.00	307.00	940.00	824.00
O_m^d	2.38	2.08	2.16	2.05	2.51
O_n^d	1.09	2.35	2.37	1.04	3.55

Table 29
The quality measures partitioning of PGP obtained for the various models.

	CPM ($k = 3$)	COPRA ($v = 11$)	GCE	EAGLE	DOCNet
Q_{ov}	–	0.39	0.23	0.21	0.23
M	–	993.00	1200.00	4924.00	1151.00
O_m^d	–	2.07	2.17	6.27	2.91
O_n^d	–	6.60	4.59	1.22	8.83

Table 30
The quality measures partitioning of Ca-CondMat network obtained for the various models.

	CPM	COPRA ($v = 11$)	GCE	EAGLE	DOCNet
Q_{ov}	–	0.25	0.28	–	0.20
M	–	1717.00	2257.00	–	1984.00
O_m^d	–	2.00	2.62	–	3.05
O_n^d	–	3.82	3.23	–	15.54

6. Conclusion

In this paper, we focus on the problem of overlapping communities detecting in social graphs which is a powerful tool for understanding the functioning of the network and its structure. Many current researches on this problem are developed and we have discussed their limits. Our main contribution is the proposition of an objective function and a local optimization algorithm called DOCNet (Detecting overlapping communities in Networks) which greedily extend a seed node until a stopping criterion is met. These proposals and their formal studies are complemented by an experimental study to compare them with the most known methods on a set of graph tests. According to these tests on artificial graphs, we found that our model detects the closest partition to the exact one and it shows a high stability especially for complex networks and when the rate of overlap between communities becomes important. In the case of real graphs, we discover that the performance of our approach is among the best. In conclusion, this work is efficient and has encouraging results. Moreover, DOCNet needs additional improvements and require further investigations. For example, our method is desirable to detect communities in noisy networks that exhibit a high number of changes over time. Another direction is interesting, too, is to adapt DOCNet model to weighted and directed networks.

References

SNAP: Stanford large network dataset collection. (2009). <http://snap.stanford.edu/data/>.

Fast and parallel implementation of eagle community detection algorithm. (2012). <http://code.google.com/p/eaglepp/>.

Benchmark graphs to test community detection algorithms. (2013). <http://sites.google.com/site/santofortunato/inthepress2>.

Greedy clique expansion. (2013). <http://sites.google.com/site/greedycliqueexpansion/>.

Network data. (2013a). <http://www-personal.umich.edu/~mejn/netdata/>.

Network data. (2013b). <http://deim.urv.cat/~aarenas/data/welcome.htm>.

Ahn, Y., Bagrow, J., & Lehmann, S. (2009). Communities and hierarchical organization of links in complex networks. *Physics Review E, 67*(2), 1–8.

Bezdek, J., Ehrlich, R., & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10*(2–3), 191–203.

Cheong, C. Y., Huynh, H. P., Lo, D., & Goh, R. S. M. (2013). Hierarchical parallel algorithm for modularity-based community detection using GPUs. In F. Wolf, B. Mohr, & D. an Mey (Eds.). *Euro-Par* (Vol. 8097, pp. 775–787). Springer.

Evans, T. S., & Lambiotte, R. (2009). Line graphs, link partitions and overlapping communities. *Physics Review E, 80*(1).

Fortunato, S. (2010). Community detection in graphes. *Physics Reports, 486*, 75–174.

Gregory, S. (2010). Finding overlapping communities in networks by label propagation. *New Journal of Physics, 12*(10), 103018.

Baumes, J., Goldberg, M., Krishnamoorthy, M., Magdon-Ismail, M., & Preston, N. (2005). Finding communities by clustering a graph into overlapping subgraphs. In *Proceedings of the IADIS international conference on applied computing* (Vol. 7, pp. 97–104).

Kim, Y., & Jeong, H. (2011). Map equation for link communities. *Physical Review E, 84*(2), 9.

Knuth, D. E. (1993). *The stanford GraphBase – a platform for combinatorial computing*. ACM.

Krebs, V. (2004). Social network of political books.

Lancichinetti, A., Fortunato, S., & KertTsz, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics, 11*(3).

Lancichinetti, A., Radicchi, F., Ramasco, José J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PLoS One, 6*(5).

Latouche, P., Birmele, E., & Ambroise, C. (2011). Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics, 5*(5), 309–336.

Lee, C., Reid, F., McDaid, A., & Hurley, N. (2010). Detecting highly overlapping community structure by greedy clique expansion. In *Fourth Workshop on Social Network Mining and Analysis SNAKDD10* (Vol. 10, p. 10).

Lo, D., Surian, D., Prasetyo, P. K., Zhang, K., & Lim, E.-P. (2013). Mining direct antagonistic communities in signed social networks. *Information Processing & Management, 49*(4), 773–791.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology, 54*(4), 396–405.

Nepusz, T., Petrczi, A., Ngyessy, L., & Bazs, F. (2008). Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E – Statistical, Nonlinear and Soft Matter Physics, 77*, 016107.

Fiumara, G., De Meo, P., Ferrara, E., & Provetti, A. (2013). Enhancing community detection using a network weighting strategy. *Information Sciences*, 648–668.

G. Palla. Clusters et communities overlapping dense groups in networks. (2011). <http://www.cfinder.org/>.

Palla, G., Derenyi, I., Farkas, I., & Vicse, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature, 435*(7043), 814–818.

Pons, P. (2004). Algorithms for large networks of interactions: Detecting community structure (Master's thesis). Paris 7 University, July 2004.

Psorakis, I., Roberts, S., & Ebden, M. (2011). Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E, 83*, 066114.

Reichardt, J., & Born holdt, S. (2004). Detecting fuzzy community structures in complex networks with a Potts model. *Physical Review Letters, 93*(21), 218701.

Shen, H., Cheng, X., Cai, K., & Hu, M. (2008). Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications, 388*(8).

Tang, J., Wang, X., & Liu, H. (2012). Integrating social media data for community detection. In M. Atzmueller, A. Chin, D. Helic, & A. Hotho (Eds.). *Proceedings of the 2011 international conference on modeling and mining ubiquitous social media* (Vol. 7472, pp. 1–20). Berlin Heidelberg: Springer.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*(6684), 409–410.

Wu, Z., Lin, Y., Wan, H., & Tian, S., (2010). A fast and reasonable method for community detection with adjustable extent of overlapping. In *IEEE international conference on intelligent systems and knowledge engineering* (pp. 376–379).

Xie, J., Kelley, S., & Szymanski, B. (2013). Overlapping community detection in networks: The state of the art and comparative study. *ACM Computing Surveys, 45*(4).

Xie, J., Szymanski, B., & Liu, X. (2011). Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Proceedings of the 2011 IEEE 11th international conference on data mining workshops* (pp. 344–349). IEEE Computer Society.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research, 33*, 452–473.

Zardi, H., & Ben Romdhane, L. (2013). An o(n2) algorithm for detecting communities of unbalanced sizes in large scale social networks. *Knowledge-Based Systems, 37*, 19–36.

Zhang, K., Lo, D., Lim, E.-P., & Prasetyo, P. K. (2013). Mining indirect antagonistic communities from social interactions. *Knowledge and Information Systems, 35*(3), 553–583.