From a Local to a Global Perspective of Community Detection in Networks

Jiamou Liu and Ziheng Wei

School of Computer and Mathematical Sciences
Auckland University of Technology, New Zealand
{jiamou.liu,bys7090}@aut.ac.nz

Abstract. We propose a novel, distributed approach for analyzing communities in social networks. In this approach, we define communities from two perspectives: local and global. Firstly, the local communities are identified by each node in a self-centred manner. Then, the global communities are captured using the notion of tendency among local communities. Our approach is especially suitable for decentralised and dynamic networks. We present formal definitions and experimentally verify our model on both static and dynamic networks.

1 Introduction

During the past 10 years, a vast amount of work has been published addressing questions involving social networks. Here, one uses graphs to represent the network topology, where the edges denote various forms of relationships or interactions among members in the network. An important attribute that differentiates social networks from an arbitrary graph is *community structure*: the distribution of edges are nonuniform in the sense that the graph contains subgraphs which are densely connected within but sparsely connected from its outside [6]. Such subgraphs are called *communities*. A central problem in this study is the detection of communities within a network (also called *clustering* in some literature).

Community detection has direct applications. For example, detecting customer communities for online retailers like Amazon helps to build effective recommendation systems [19]. Detecting scientific communities based on research collaboration networks helps to reveal collaboration patterns [15] among research fields. Detecting communities in the protein-protein interaction (PPI) networks help to reveal functional groups that are associated to cancer and metastasis [9]. Other applications appeared in ecology [11], epidemiology [13] and counterterrorism [25].

Community detection also proves to be hard: 1) Due to the wide varieties of networks, it is extremely difficult to provide a formal definition of communities. Numerous definitions have been proposed, yet no definition has been universally accepted. 2) Communities are not homogeneous; large but relatively sparse communities may contain smaller and denser sub-communities. Hence one needs notions and tools for dealing with relations among communities. 3) One needs

D.-N. Pham and S.-B. Park (Eds.): PRICAI 2014, LNAI 8862, pp. 1036–1049, 2014.

feasible computation methods for detecting communities. Unfortunately many existing definitions of communities result in high computation costs for community detection (e.g., two well-known approaches – one based on modularity optimization [14] and another based on clique percolation [17] – both lead to NP-hardness of the community detection problem).

More recently, two trends in network analysis emerge which pose new challenges: i) As networks grow larger, data become increasing *decentralised* (e.g. the Internet). One needs new techniques for handling networks when only *local information* is present at each node. ii) Real networks are often *dynamic* in the sense that it undergoes continuous changes over time (e.g. online social network, phone-call network). Thus communities are no longer static; they emerge and evolve in a dynamic manner. Therefore one needs computational methods that keep track of *community evolution* in rapidly changing networks.

Our Contribution. We tackle the challenges above by proposing a new model of communities. Our motivation is the following: one may view the communities in a network from two perspectives. From a *local perspective*, the community of a node v consists of those nodes with whom v would agree to associate. From a *global perspective*, local communities may "tend" to combine in order to optimize certain global criteria. Thus our community model includes:

(1) *Local communities*: Intuitively speaking, a local community of a node v is the "inner ring" of v, which is a "tightly-knitted" cluster of nodes with very high density. The local communities are detected through optimizing a certain *utility* measure of the clusters. This is in principle similar to the approach adopted by e.g. [10]. (see Sec. 4)

(2) *Global communities*: Several local communities may be so closely linked that they tend to form a single community, even though combining them would not improve the overall utility. We capture this situation using the notion of *tendency*, which helps to detect communities from a global perspective. Through numerous examples, we stipulate that such tendency among local communities is a general phenomenon in real-life communities. (see Sec. 5)

To make our approach useful in decentralised and dynamic network, we make the following assumptions for community detection:

A1 : Each community only sees local information regarding its neighbourhood. There is no common knowledge shared among all nodes.

A2 : Each community makes self-centred, subjective decisions based on its information. No central controller exists in the network.

We implement and test our community detection algorithm on both real and generated data. Firstly we apply our model to three well-known examples: (1) Zachary's karate club, (2) bottlenose dolphine in Doubtful Sound, and (3) American college football league. These experiments show that our model captures real-life communities with high precision. We then use generated data to simulate community detection in dynamic networks.

Related Work. Numerous community detection methods have been proposed in the past. The reader is referred to [6,14,1,17,3,4,21,18] for important breakthroughs and surveys to this field. The recent survey [7] gave a state-of-the-art introduction on clustering dynamic networks. Despite intensive research in community detection, few distributed methods have been proposed; see [24,23,5,26] for works in this direction. Lastly, we mention Massaro et al's recent work on localised community detection using a probabilistic model [12].

Paper Organization. Section 2 presents required terminology. Section 3 presents the framework for our distributed community detection algorithm. Section 4 and Section 5 present our model of local and global communities. Section 6 and Section 7 discuss experiments on both real static networks and generated dynamic networks. Section 8 concludes the paper.

2 Preliminaries

We represent a network as an unweighted undirected graph where every node is an element of the network.Girvan and Newman discovered that very often real networks exhibit community structure: nodes in the network can be partitioned into clusters, with high density of edges within each cluster, but low density of edges between these clusters [6]. Our goal is thus to identify such clusters in a community structure, using only information provided by the graph topology.

A *graph* G is a pair (V, E) where V is a set of nodes and $E \subseteq \{\{u, v\} \mid u \neq v \in V\}$ is a set of undirected edges on V. We abuse the notation writing an undirected edge as (u, v) instead of $\{u, v\}$. For any set S, $|S|$ denotes the cardinality of S. A *neighbour* of a node v is any node u with $(v, u) \in E$. For any node $v \in V$, the *neighbourhood* of v is the set $N(v) = \{v\} \cup \{u \in V \mid (v, u) \in E\}$ that consists v and all its neighbours. For $C \subseteq V$, set $N(C) = \bigcup_{u \in C} N(u)$. A graph is *connected* if there is a path between any pair of nodes.

Definition 1. *A* clustering *of a graph $G = (V, E)$ is a collection of sets $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ where each $C_i \subseteq V$ induces a connected subgraph of G, $C_i \cap C_j = \varnothing$ for any distinct i, j, and $\bigcup_{i=1}^{k} C_i = V$. Each C_i is called a* cluster *in $\mathcal{C}$. A set $C \subseteq V$ is said to be $\mathcal{C}$-*consistent *if it is the union of sets in $\mathcal{C}$.*

Intuitively speaking, a *community* in a graph $G = (V, E)$ is a connected induced subgraph that has high intra-cluster density but low inter-cluster density. Let C be a set of nodes. The intra-cluster density of C measure the edge connectivity of the subgraph induced on C:

Definition 2. *The* intra-cluster density $\delta_{\mathsf{int}}(C)$ *of C is the ratio between the actual number of edges in C and the total number of possible edges in C, i.e.,*

$$\delta_{\mathsf{int}}(C) = \begin{cases} \frac{|C^2 \cap E|}{|C|(|C|-1)/2}, & \text{if } |C| > 1, \\ 1, & \text{if } |C| = 1. \end{cases}$$

Let $\mathcal{C}$ be a clustering of $G = (V, E)$. Take a set C that is $\mathcal{C}$-consistent. The inter-cluster density of C with respect to $\mathcal{C}$ measures the connectivity between C and its neighbouring clusters:

Definition 3. *Let $D_1, D_2, \ldots, D_m$ be all clusters in $\mathcal{C}$ disjoint from C that contain a neighbour of C, i.e., $C \cap D_i = \varnothing$, $D_i \cap N(C) \neq \varnothing$ for all $1 \leq i \leq m$. The* inter-cluster density *of C with respect to $\mathcal{C}$ is defined as*

$$\delta_{\mathsf{ext}}(C, \mathcal{C}) = \begin{cases} \frac{|\{(v,u)\in E \mid v\in C, u\in N(C)\setminus C\}|}{|C|\times(|D_1|+|D_2|+\cdots+|D_m|)}, & \text{if } m \geq 1 \\ 0, & \text{if } m = 0. \end{cases}$$

Def. 3 is different from the typical definition of inter-cluster density (such as the one given in [4]), which measures connectivity between C and the rest of the entire network. The reason for this is our assumption A1, which states that a node is only aware of its neighbourhood and its community, and hence would not be able to grasp the entire network.

3 An Algorithmic Framework

Instead of describing a central controller, we specify the actions of individual computational units, called *cells*, in a number of *rounds* of computation:

- **Round** 1**:** Each node is a cell. The goal is to compute a partition $\mathcal{C}_1$ where each cluster in $\mathcal{C}_1$ is called an *initial cluster.*
- **Round** $i > 1$**:** Each cluster in $\mathcal{C}_{i-1}$ is a cell. The goal is to compute a clustering $\mathcal{C}_i$ where each cluster in $\mathcal{C}_i$ is $\mathcal{C}_{i-1}$-consistent. The sequence of clusterings $\mathcal{C}_1, \mathcal{C}_2, \ldots$ eventually stablises and we call the limit clustering $\mathcal{C}_{\mathsf{loc}}$, in which each cluster is called a *local community.*
- **Round** ω**:** Each local community is a cell. The goal is to compute the clustering $\mathcal{C}_{\mathsf{glo}}$ where each cluster in $\mathcal{C}_{\mathsf{glo}}$ is a *global community.*

In each round, a cell C performs three steps: Invite, Select and Join:

- Invite: The cell C evaluates its neighbouring cells and based on the result sends out "invitations" to a subset $I(C)$ of its neighbours.
- Select: The cell C evaluates the utility of all invitations it receives, and selects among the senders a cell $s(C)$.
- Join: The cell C joins the cluster proposed by its selected cell $s(C)$.

See Fig. 1 for a flow chart. In subsequent sections we present formal definitions of our community models, which naturally provide implementations of the Invite, Select and Join steps in each round.

4 Local Communities

Cliques are maximally connected subgraphs (where there is an edge between any pair of nodes) in the graph. It is thus desirable to define the initial cluster $C_1(v)$ as the maximal cliques that contains v. However, finding maximal cliques is a well-known NP-hard problem. The problem is even hard for approximation in a precise sense [8]. We therefore use a different notion to obtain a *clique-like* subgraph from the neighbourhood $N(v)$. The definition below is a slight variation of the notion introduced in [22].

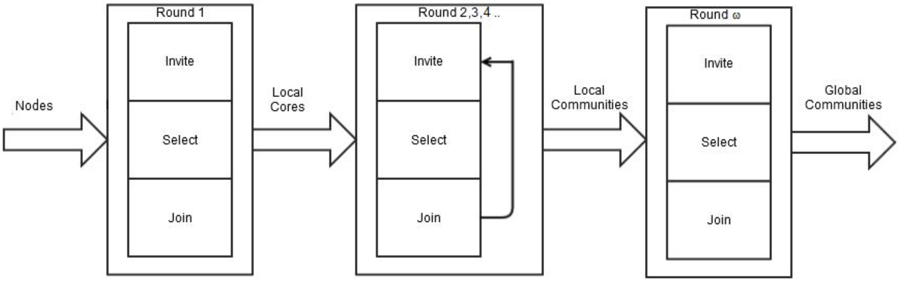

Fig. 1. The flow of our algorithmic framework

Definition 4. *A* k-core *in a graph is an induced subgraph where all nodes have degree at least* k *. The* core number *of a node* v *is the largest* $\kappa(v)$ *such that* $N(v)$ *contains a* $\kappa(v)$*-core. For* $v \in V$*, the* local core *of* v *is the set*

$$K(v) = \{u \in N(v) \mid |N(u) \cap N(v)| \geq \kappa(v) - 1\}$$

See Figure 2 for an example of the local core of nodes. Note that in the example $K(v)$ of any v coincides with the maximal clique containing v. Furthermore, the local core $K(v)$ of a node v necessarily contains the node v, and hence is connected. See Appendix A for an efficient algorithm that computes $K(v)$.

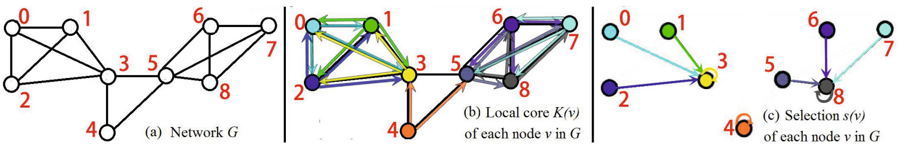

Fig. 2. (a) An example network G. (b) Nodes in $K(v)$ are represented by the outgoing arrows from v. Nodes in $I(v)$ are represented by incoming arrows to v. (c) The arrow from a node v indicates the selection $s(v)$. This results in the initial clustering $\mathcal{C}_1 = \{\{0, 1, 2, 3\}, \{5, 6, 7, 8\}, \{4\}\}$.

The local communities essentially optimize the following community utility.

Definition 5. *Let* $\mathcal{C}$ *be a clustering of the graph* G*, and* $C_1, C_2 \subseteq V$ *be* $\mathcal{C}$*-consistent. We say that* C_2 *has higher* community utility *than* C_1*, written as* $C_1 \prec C_2$*, if one of the following holds:*

- $\delta_{\mathsf{int}}(C_1) < \delta_{\mathsf{int}}(C_2)$*;*
- $\delta_{\mathsf{int}}(C_1) = \delta_{\mathsf{int}}(C_2)$ *and* $|C_1| < |C_2|$*;*
- $\delta_{\mathsf{int}}(C_1) = \delta_{\mathsf{int}}(C_2)$*,* $|C_1| = |C_2|$ *and* $\delta_{\mathsf{ext}}(C_1, \mathcal{C}) > \delta_{\mathsf{ext}}(C_2, \mathcal{C})$

Round 1. When we compare the community utilities in Round 1, we use the basic clustering $\mathcal{C}_0$ where each node forms a cluster itself. Intuitively speaking, each clustering $\mathcal{C}_1$ is a two-way selection process: Every node v first sends invitations to nodes in its local core $K(v)$. Upon receiving an invitation, v then evaluates the community utility of the proposed local core, and selects the node w whose proposal has the highest utility.

Formally, for each $v \in V$, let $I(v) = K(v)$. Let $R(v) = \{u \in N(v) \mid v \in I(u)\}$; these are all nodes that v receives invitations from. The *selection* $s(v)$ is the node $w \in R(v)$ that has the highest community utility. Define the set $\mathsf{Sel}(v) = \{u \in N(v) \mid s(u) = v\}$; these are all nodes who have selected v.

Definition 6. *The* initial cluster $C_1(v)$ *of a node v is the connected component of v in the subgraph induced by* $\mathsf{Sel}(s(v))$*. The* initial clustering $\mathcal{C}_1$ *is the collection of all initial clusters in the graph.*

See Fig. 2(c) for the selections and initial clusters in the previous example.

Round $i > 1$. Assume that we have defined $\mathcal{C}_{i-1}$ for $i > 1$. Our next goal is to define a new clustering $\mathcal{C}_i$. This is similar to the initial clustering step: each cluster $C \in \mathcal{C}_{i-1}$ chooses a cluster $D \in \mathcal{C}_{i-1}$ (which may be C itself) from its neighbourhood based on the community utility of $C \cup D$ and sends an invitation to D. The cluster D then selects among all received invitations one that maximizes the community utility.

Formally, define for every cluster $C \in \mathcal{C}_{i-1}$ its *periphery* $\rho(C) = \{D \in \mathcal{C}_{i-1} \mid D \cap K(v) \neq \varnothing$ for some $v \in C\}$ which consists of all clusters in $\mathcal{C}_{i-1}$ that contain a node in the local core of some node in C. Set $I(C)$ as a singleton $\{D\}$ where $D \in \rho(C)$ and $C \cup D$ has maximal community utility with respect to the clustering $\mathcal{C}_{i-1}$. Note that implicitly $C \in I(C)$ and hence it may also send an invitation to itself. Let $R(C) = \{D \mid C \in I(D)\}$. The selection $s(C)$ is defined as the community $D \in R(C)$ such that $C \cup D$ has the highest community utility. Note that whenever $s(C) \neq C$, $s(C)$ always has higher community utility than C. Hence for every $C \in \mathcal{C}_{i-1}$, there is some $j \in \mathbb{N}$ and $D \in \mathcal{C}_{i-1}$ such that $D = s^j(C) = s^{j+1}(C)$. We call D the *i-sink* of C.

Definition 7. *Two clusters $C_1, C_2 \in \mathcal{C}_{i-1}$ belong to the same cluster in $\mathcal{C}_i$ if they have the same i-sink.*

For $i \in \mathbb{N}$ and $v \in V$ we use $C_i(v)$ to denote the cluster in $\mathcal{C}_i$ that contains v. As the network G is finite, the sequence $C_1(v), C_2(v), \ldots$ eventually stablises.

Definition 8 (Local Communities). *Let $G = (V, E)$ be a network. The* local community $C_{\mathsf{loc}}(v)$ *of a node $v \in V$ is defined as $C_i(v)$ where i is the least number such that $C_i(v) = C_{i+1}(v)$.*

We use two examples to illustrate the notion of local communities. The first example (Fig. 3(a)) is the graph shown in Fig. 2(a), which stablises after the initial step. The second example (Fig. 3(b)) identifies four local communities based on the above definitions after two iterations.

5 Global Communities

The local community $C_{\mathsf{loc}}(v)$ has, in a certain sense, reached *local optimality* – it cannot be combined with other local communities which leads to a higher intra-cluster density, nor can it achieve a lower inter-cluster density without

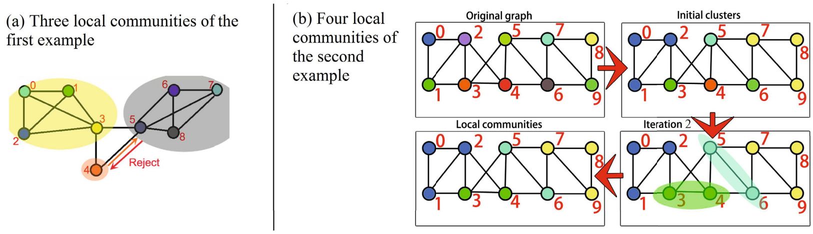

Fig. 3. (a) The cluster $\{4\}$ sends one invitation to $\{5, 6, 7, 8\}$, but is not selected by these nodes. Hence no change is made to the initial clustering. We have $\mathcal{C}_2 = \mathcal{C}_1 = \{\{0, 1, 2, 3\}, \{5, 6, 7, 8\}, \{4\}\}$ and they are the local communities. (b) Clusters are shown as nodes with the same colour. The resulting local communities are $\mathcal{C}_2 = \{\{0, 1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8, 9\}\}$.

sacrificing its intra-cluster density. Such clustering reveals important properties regarding connectivity of individual nodes. However, in real-world networks that are sparse (e.g. terrorist networks), the local communities tend to be very small. The granularity of the local community clustering $\mathcal{C}_{\mathsf{loc}}$ is thus too fine to reveal any significant global structural property in practice.

It is important to note that two local communities may still be closely linked, even though their combination does not result in a better community utility. Thus, seeing from a global perspective, it may be reasonable to combine them. Here we introduce the notion of *tendency*: intuitively, when one local community has a lower community utility than a neighbouring one, this community is "attracted" towards the other community. Recall that $K(v)$ denotes the local core of a node $v \in V$. Formally, we define the following:

Definition 9. *Let $C \in \mathcal{C}_{\mathsf{loc}}$ be a local community. The* periphery *of C is the set $\rho(C) \subseteq \mathcal{C}_{\mathsf{loc}}$ defined as $\rho(C) = \{D \in \mathcal{C}_{\mathsf{loc}} \mid D \cap K(v) \neq \varnothing$ for some node $v \in C\}$. We say that C has a* tendency *towards D if $D \in \rho(C)$ and D has the highest community utility among all local communities in $\rho(C)$ with respect to the local community clustering $\mathcal{C}_{\mathsf{loc}}$; in this case, we write $\mathsf{Td}(C) = D$.*

Note that the above definition implicitly implies that a local community always belongs to its own periphery, and hence it may have a tendency towards itself. As each local community has tendency towards exactly one local community, the tendencies of all local communities in $\mathcal{C}_{\mathsf{loc}}$ form a forest:

Definition 10. *The* tendency forest *$\mathcal{F}_{\mathsf{Td}}$ of a network G is a directed graph whose nodes are all local communities $\mathcal{C}_{\mathsf{loc}}$ and there is an edge (C, D) in $\mathcal{F}_{\mathsf{Td}}$ if and only if $D = \mathsf{Td}(C)$.*

The *tendency sink* of a local community $C \in \mathcal{C}_{\mathsf{loc}}$ is the root of the tree that contains C in the tendency forest $\mathcal{F}_{\mathsf{Td}}$, i.e., it is the local community $D = \mathsf{Td}^i(C)$ for some $i \geq 1$ such that $\mathsf{Td}^i(C) = \mathsf{Td}^{i+1}(C)$. The above notions allow us to define global communities in a network:

Definition 11 (Global Communities). *Let $G = (V, E)$ be a network. The* global community *of a node $v \in V$ is the set of all $u \in V$ such that the local communities $C_{\mathsf{loc}}(v)$ and $C_{\mathsf{loc}}(u)$ have the same tendency sink.*

Note that tendency sinks can also be viewed as a centrality notion in the graph, which represent a dense cluster that has the greatest "influence" in a community. As an example, we show in Fig. 4 the tendency forest of the graphs in Fig. 3.

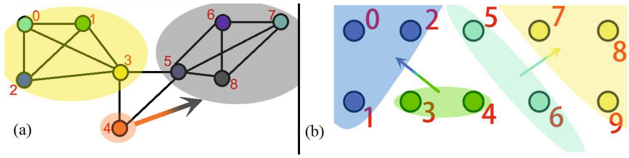

Fig. 4. (a) The tendency forest of the network shown in Fig. 3(a). Each shaded area represents one local community. The arrows represent the tendency among local communities. The global communities are $\{0, 1, 2, 3\}$ and $\{4, 5, 6, 7, 8\}$. (b) The tendency forest of the network shown in Fig. 3(b). Each shaded area represents one local community. The arrows represent the tendency among local communities. The global communities are $\{0, 1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$.

Computation Complexity. we remark on the efficiency of detecting global communities in a network. The algorithm is very efficient: the overall time cost on each node $v \in V$ is polynomial. Indeed, our algorithm for computing the local core of v takes time $O(n_v^2)$ where $n_v = |N(v)|$. The running time of each round i is $O(|E|)$ and there are at most $O(|V|)$ rounds. Given the local communities, computing global communities requires $O(m)$ time in total. Furthermore, the notions of local cores, local and global communities uses only local information, hence the computation can be distributed among nodes and local communities, where information exchanges happen only among neighbouring cells. Implementing the algorithm in a distributed network would drastically improve computation time when the network becomes large.

6 Experiments: Static Community Detection

We implemented our algorithms using the Java programming language with OpenGL for visualization. The implementation is distributed: each cluster executes a thread, which repeatedly performs the three steps Invite, Select and Join until it joins another cluster. We perform experiments on a laptop with Intel Core i7-3630QM CPU 2.4GHz 8.0GB RAM. The experiments involve three well-known benchmarks in the *static case*. The results of the following experiments are recorded in videos which were uploaded to the indicated URLs.

6.1 Zachary's Karate Club (Video: `http://youtu.be/dNimOto3QSU`)

Zachary models the friendship among all 34 members of a karate club in an American university. After an internal dispute, the club split into two factions: one centered on 0 (the instructor's faction), and another one centered on node 32 and 33 (the officer's faction). Zachary correctly predicated through analysing the graph how the club is to be split prior to the actual split [27]. The example has since become an ubiquitous case study taken by many researchers.

After running our algorithm on Zachary's network we obtained two global communities (see Fig. 5), which match exactly with the factions. The local communities identified by our algorithm reveal a finer-grained association among the members, and closely resemble the partition found using the modularity-based method [16] and the clique-based method [3]. In particular, two large local communities are located: $\{0, 1, 2, 3, 13\}$ and $\{8, 30, 32, 33\}$, which are the tendency sinks of the two global communities. They coincide well with the known centers of the two factions: 0 is the instructor, and 33 is the president.

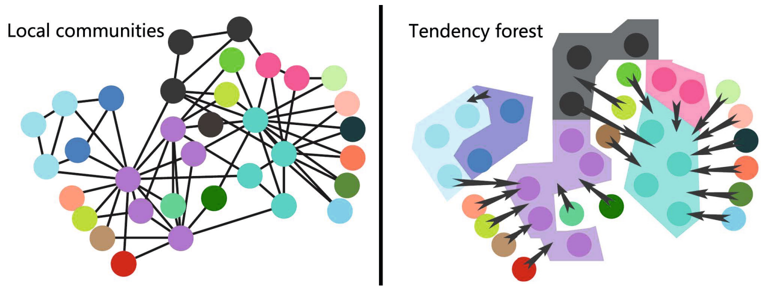

Fig. 5. Zachary's karate club graph with 34 nodes. Left: Two nodes belong to the same local communities if they have the same colour. Right: An arrow represent the tendency from one local community to another.

6.2 Bottlenose Dolphin Network (Video `http://youtu.be/kAAiYn5PN_0`)

The dolphin social network studied by Lusseau captures interactions among bottlenode dolphines in Doubtful sound in New Zealand. One of the classic research [11] on animal social networks applied Girvan-Newman algorithm [6] to the dolphin social network and detected four communities. The detection required to use extra information such as gender of dolphins in their experiments. These extra information actually played a significant role in finding these natural communities. We conducted experiments on the same network excluding those extra information. Our program identified four communities same as the results obtained using the Girvan-Newman algorithm. The local communities and the tendency forest are shown in Fig. 6. The resulting global communities misplaced only two dolphins compared to the result in [11]. We also include a table showing the intra-/inter-community density of each community; the intra-community density is roughly 4-7 times of the inter-community density.

6.3 American Football League (Video: `http://youtu.be/CUPFl2ECECs`)

The American college football league network captures matches between teams. The league consisted of several conferences which are normally classified by

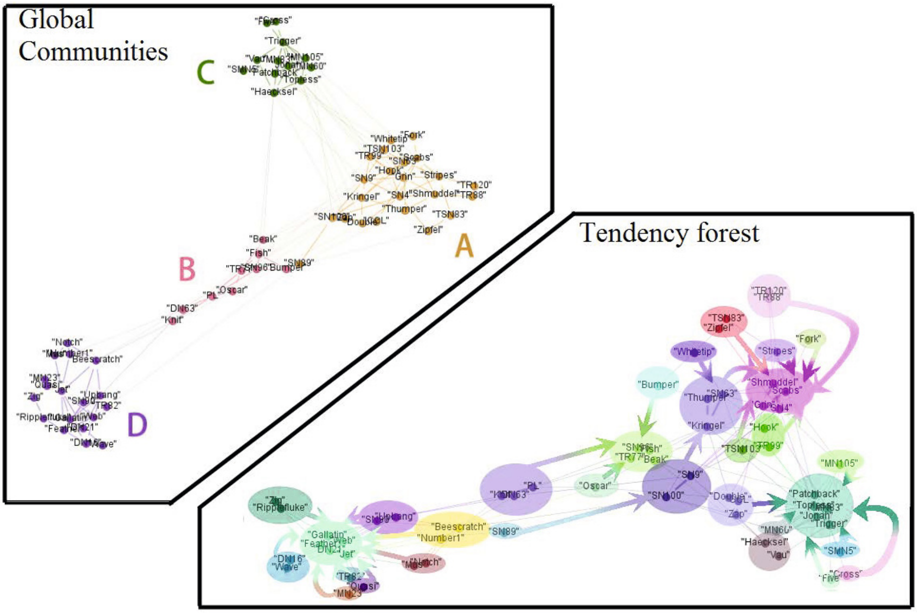

Fig. 6. The dolphin social network. Left: The global communities. Right: The local communities and the tendency forest

their geographic locations. Teams within the same conference interact more than teams between different conferences. Sometimes teams at intersectional region interact more with teams outside of their own conferences. The league has 12 conferences in total but our system partitions one of the conferences into two. It can be statistically justified that the disrupt conference is located at a centric area of America. In this experiment, we found that the tendency between two local communities implicitly reveals the idea of overlapping community. Tendency only appears when two local communities have proposed to the same set of candidates before.

7 Experiment: Dynamic Network Detection

To illustrate the use of our distributed algorithm in the dynamic setting, we simulated networks that undergo dynamic changes. Firstly we implement a network with 60 nodes based on Condon and Karp's planted ℓ-model where $\ell = 4$ [2]. Edges are added to the network in sequence so that the graph contains four subgraphs containing more edges within each subgraph (with probability 0.9) and less edges between subgraphs (with probability 0.1). The algorithm correctly identified all four planted communities. See the video (see `http://youtu.be/43cW9CSbg10`).

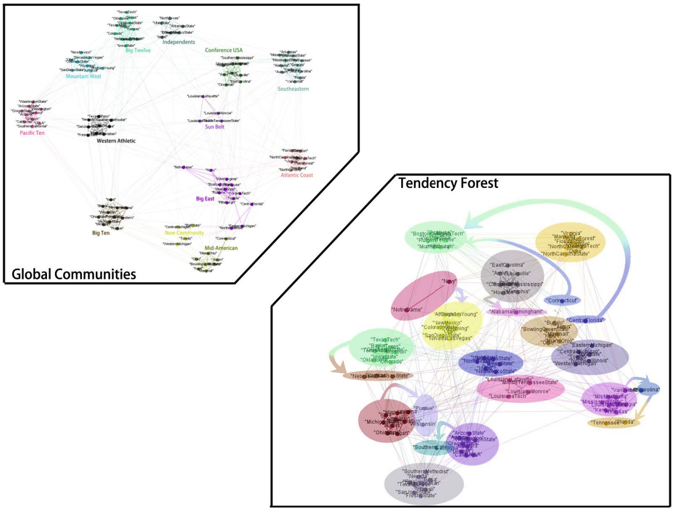

Fig. 7. The American colledge football league network. Left: The global communities. Right: The local communities and the tendency forest.

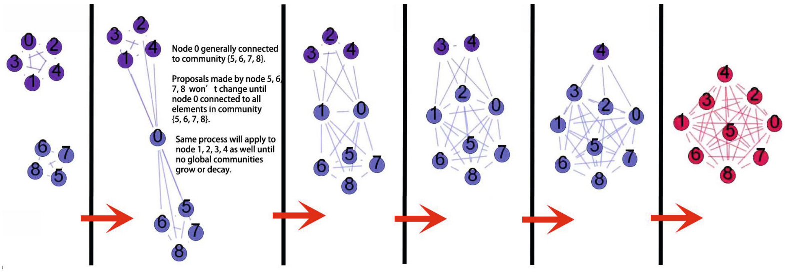

Fig. 8. Edges were built between two strongly connected component

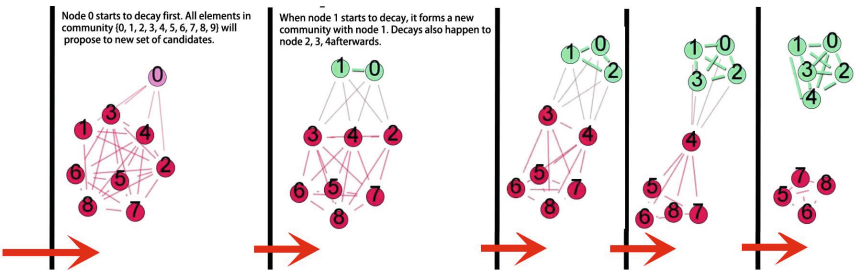

Fig. 9. Remove edges that have been added between two strongly connected components and re-identify original components

Secondly we implemented a network with 75 nodes containing four strongly connected components that are complete graphs. Edges are added in sequence between these components so that eventually the nodes merge to a single components hence forming one community. Then edges are removed in sequence so that in the end four communities re-emerge in the network. See the video in `http://youtu.be/cdUcBzrVjcA`. This video illustrated how global communities evolved during the sequence of changes.

We also show using Fig. 8 and Fig. 9 another smaller example that illustrates how two global communities grow towards one and disintegrate from one into two global communities again. The underlying mechanism is that a change made to a node may affects the entire initial cluster which it belongs to. Moreover, the other nodes in the same initial cluster will re-propose regarding changes in order to form new initial cluster.

8 Discussion and Conlusion

In this paper we proposed a formal model of communities from the local and global perspectives and implement an algorithm to realise this model. Our approach is efficient and captures communities in dynamic networks in a decentralised way. Communities identified by our algorithm almost conform to communities in every real network that we have used in our experiments. Since these networks vary significantly, our definition to community has certain degrees of generality. To further justify the validity of our results, we computed modularity of the identified communities (with different rewire probability), which is a well-known performance measure for community detection algorithms [4]. The following table shows that our results achieve almost the same modularity as the results obtained by a well-known modularity optimisation algorithm [1].

Rewiring probability	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13
Our approach	0.8108	0.8065	0.8015	0.7773	0.7758	0.7759	0.7546	0.7476	0.7271
Compared approach	0.8108	0.8065	0.8015	0.7773	0.7758	0.7759	0.7547	0.7476	0.7270

Rewiring probability	0.14	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22
Our approach	0.7252	0.7167	0.7250	0.7086	0.6934	0.6715	0.6384	0.66150	0.6143
Compared approach	0.7269	0.7167	0.7250	0.7086	0.6934	0.6765	0.6653	0.6615	0.6415

As future work, we are still challenged by large datasets (hardware limitation) and the complication of collecting real life dynamic network datasets. The mentioned distributed algorithm was implemented in a single-core computer running multiple thread. Therefore the experiment did not reveal the efficiency of the algorithm and does not scale to huge networks. We expect that the algorithm will be able to scale to very large networks once we implemented it in a multi-core computer in a truly distributed manner.

References

1. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of community hierarchies in large networks. Journal of Statistical Mechanics (10), P10008 (2008)
2. Condon, A., Karp, R.: Algorithms for Graph Partitioning on the Planted Partition Model. Random Struct. Algor. 18, 116–140 (2001)
3. Evans, T.: Clique graphs and overlapping communities. Journal of Statistical Mechanics: Theory and Experiment 2010, P12037 (2010)
4. Fortunato, S.:: Community detection in graphs. CoRR abs/0906.0612 (2010)
5. Gregori, E., Lenzini, L., Mainardi, S.: Parallel k-clique community detection on large-scale networks. IEEE Transactions of Parallel and Distributed Systems 24(8), 1651–1660 (2013)
6. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the USA 99(12), 7821–7826 (2002)
7. Harmann, T., Kappes, A., Wagner, D.: Clustering evolving networks. CoRR abs/1401.3516 (2014)
8. Håstad, J.: Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Mathematica 182(1), 105–142 (1999)
9. Jonsson, P., Cavanna, T., Zicha, D., Bates, P.: Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. BMC Bioinformatics 7, 2 (2006)
10. Kim, S.: Graph theoretic sequence clustering algorithms and their applications to genome comparison. In: Computational Biology and Genome Informatics, pp. 81–116. World Scientific Publishing Company (2003)
11. Lusseau, D., Newman, M.: Identifying the role that individual animals play in their social network. Proceedings of the Royal Society of London. Series B: Biological Sciences 271(6), 477–481 (2004)
12. Massaro, E., Olsson, H., Guazzini, A., Bagnoli, F.: Impact of local information in growing networks. In: Proc. of Wivace 2013 – Italian Workshop on Artificial Life and Evolutionary Computation. EPTCS, vol. 130, pp. 53–60 (2013)
13. Meyers, L., Newman, M., Martin, M., Schrag, S.: Applying network theory to epidemics: Control measures for outbreaks of mycoplasma pneumoniae. Emerging Infectious Diseases 9(2), 204–210 (2003)
14. Newman, M.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133 (2004)
15. Newman, M.: Coauthorship networks and patterns of scientific collaboration. Proc. Natl. Acad. Sci. USA 101, 5200–5205 (2004)
16. Newman, M.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103, 8577–8582 (2006)
17. Palla, G., Der'enyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
18. Porter, M., Onnela, J.-P., Mucha, P.: Communities in networks. Notices of the AMS 56(9), 1082–1097 (2009)
19. Reddy, P.K., Kitsuregawa, M., Sreekanth, P., Rao, S.S.: A graph based approach to extract a neighborhood customer community for collaborative filtering. In: Bhalla, S. (ed.) DNIS 2002. LNCS, vol. 2544, pp. 188–200. Springer, Heidelberg (2002)
20. Rossi, R., Gleich, D., Patwary, M.: Parallel maximum clique algorithms with applications to network analysis and storage. CoRR abs/1302.6256 (2013)

21. Schaeffer, S.: Graph clustering. Computer Science Review 1(1), 27–64 (2007)
22. Seidman, S.: Network structure and minimum degree. Social Networks 5(3), 269–287 (1983)
23. Staudt, C., Meyerhenke, H.: Engineering High-Performance Community Detection Heuristics for Massive Graphs. In: Proceedings of the 2013 International Conference on Parallel Processing, pp. 180–189. IEEE (2013)
24. Shi, J., Xue, W., Wang, W., Zhang, Y., Yang, B., Li, J.: Scalable community detection in massive social networks using MapReduce. IBM Journal of Research and Development 57(3/4), 12 (2013)
25. Weinstein, C., Campbell, W., Delaney, B., O'Leary, G.: Modeling and Detection Techniques for Counter-Terror Social Network Analysis and Intent Recognition. In: Proc. of IEEE Aerospace Conference, pp. 1–16 (2009)
26. Yildiz, H., Kruege, C.: Detecting social cliques for automated privacy control in online social networks. In: Proc. of Pervasive Computing and Communications Workshops, pp. 353–359. IEEE (2012)
27. Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropoloical Research (33), 452–473 (1977)