如何写应用案例（4.3）
小例子 图1

A Game-Theoretic Framework to Identify Overlapping Communities in Social Networks

Wei Chen[1], Zhenming Liu[2], Xiaorui Sun[3], and Yajun Wang[1]

[1] Microsoft Research Asia, Beijing, China. weic,yajunw@microsoft.com
[2] Harvard School of Engineering and Applied Sciences. zliu@fas.harvard.edu
[3] Department of Computer Science, Shanghai Jiao Tong University.
sunsirius@sjtu.edu.cn

Abstract. In this paper, we introduce a game-theoretic framework to address the community detection problem based on the structures of social networks. We formulate the dynamics of community formation as a strategic game called *community formation game*: Given an underlying social graph, we assume that each node is a selfish agent who selects communities to join or leave based on her own utility measurement. A community structure can be interpreted as an equilibrium of this game.

We formulate the agents' utility by the combination of a gain function and a loss function. We allow each agents to select multiple communities, which naturally captures the concept of "overlapping communities". We propose a gain function based on the modularity concept introduced by Newman (2006), and a simple loss function that reflects the intrinsic costs incurred when people join the communities. We conduct extensive experiments under this framework, and our results show that our algorithm is effective in identifying overlapping communities, and are often better then other algorithms we evaluated especially when many people belong to multiple communities.

To the best of our knowledge, this is the first time the community detection problem is addressed by a game-theoretic framework that considers community formation as the result of individual agents' rational behaviors.

1 Introduction

Understanding the formation and evolution of communities is a long-standing research topic in sociology because in part of its important relations with the studies of urban development [9], criminology [22], social marketing [13, 10], and many other areas. With recent increasing popularity of online social network services like Facebook and Twitter, the studies of community structures become more important than ever. Among these studies, identifying and detecting the communities are not only of particular prominence but have immediate applications. For example, for effective online marketing, such as placing online ads or deploying viral marketing strategies, identifying communities in the social network could often lead to more accurate targeting and achieve better marketing results. Albeit online user profiles or other semantic information is certainly helpful to identify user segments and communities, this kind of information is often at a coarse-grained level and overlooks information on communities at a more fine-grained

level. On the other hand, the structure of social networks reveals rich information over the community structure, especially at the fine-grained level. In this paper, we study community detection based on the network structure.

There exists extensive research work on detecting communities sheerly based on network structures, for which we provide a more detailed introduction in the next section. From these researches, however, we find a couple of issues that we feel are less satisfactory. First, many community detection algorithms either set a global optimization goal for detection, such as optimizing for *modularity* [17], *betweenness* [8], or *conductance* [3], or has no optimization goal and just looking for sub-components with certain predetermined structures (e.g. [21]). Neither of them is grounded with a systematic theory for the emergence of communities over a social network. Communities in real social networks are certainly formed to serve a purpose, but they are also formed organically from bottom up, without a global authority trying to enforce a global optimal objective. Second, a majority of the researches in community detection focus on partitioning graphs into disjoint components and only a few of them accommodate overlapping communities. In social networks, every individual typically belongs to more than one communities, such as the community of her family members, that of her friends and classmates, that of her co-workers, etc. Therefore, a satisfactory community detection algorithm shall incorporate overlapping communities, and preferably do so in a natural way.

In this paper, we address the above two issues by borrowing game-theoretic concepts often used in economic researches. We model each node in a social network as a rational agent trying to optimize its own utility by joining or leaving communities, which we define as a *community formation game*. A Nash equilibrium of the game can be readily interpreted into a community structure of the network: the communities every node belongs to in a Nash equilibrium become the output of our community detection algorithm. Due to computational constraints, we use local equilibria [1] instead of Nash equilibria as our solutions. We believe that this game-theoretic framework reflects real-world organic community formations, and thus in principle is a more systematic framework than others that rely on global optimization objectives or have no objectives. This framework also naturally incorporates overlapping communities, because we allow each individual agent to select multiple communities, as long as it could improve her utility.

In our framework, the utility of each individual consists of two parts: a gain function and a loss function. This is to match the real-world scenario, in which each individual not only receives benefit from the communities she belongs to but also needs to pay certain cost to maintain her membership in the communities. To materialize our framework into actual community detection algorithms, we address a number of issues, in particular the existence and the computation efficiency of Nash equilibria in our community formation game.

For the existence of Nash equilibria, we show that in general Nash equilibria may not exist, but we provide a sufficient condition to permit the existence of Nash equilibria. More specifically, we define a family of functions called locally linear functions and show that if all gain and loss functions of all the agents are locally linear, the game is guaranteed to have a Nash equilibrium. For computation efficiency, we show that some locally linear gain and loss functions will result in games for which the computation of

Nash equilibria belongs to hard computation classes such as NP-hard or PLS-complete classes.

To deal with these computational issues, rather than computing a Nash equilibrium of a game, we focus on finding *local equilibria* [1], which are the states such that no agent can improve her own utility with a "small" deviation from her current strategy. Notice that the Nash equilibria are local equilibria by definition. We are able to define a set of gain and loss functions that are both locally linear and allow efficient computations of the *local equilibria*, when the agents' strategy space is locally constrained.

Our gain function is based on a "personalized" version of the modularity [17], and thus reflects the individual desire of forming a close-knit community structure. Our proposed loss functions are based on the number of communities one need to maintain.

We conduct experiments on both synthetic and real networks to evaluate our community detection algorithm. It is not easy to find social networks with an associated underlying communities as a ground truth, and thus we adopt the recently proposed synthetic benchmark graphs of [11] to quantitatively evaluate our algorithm and compare with two recently proposed detection algorithms that allow overlapping communities: the clique percolation method [21] and the CONGA algorithm [8]. Our results show that our algorithms performs quite well overall, and are usually better than the other algorithms especially when more nodes belong to multiple communities. We also run our algorithms on a couple of real networks, and show visually the results of our detection. Finally, as a proof of concept application, our algorithm is used to disambiguate Chinese authors with the same English names in DBLP and our results look promising.

As a summary, our contribution is to propose the use of a game-theoretic framework for the community detection problem and to provide an actual algorithm based on this framework. Our experimental results demonstrate that our algorithm is very effective, especially when dealing with the case where a large portion of people belong to multiple communities. To the best of our knowledge, this is the first work that adopts a game-theoretic approach in community detection problem. We believe that this approach more naturally reflects real-world community formation in social networks and is a promising direction worth further investigation and expansion.

1.1 Related work

The earliest result related to community detection in computer science is probably the graph partition problem, which can date back to the design of VLSI [6] and modeling roles and positions in social structure [24]. While these approaches are relevant to the community detection, Newman pointed out a few facts that make the approaches unsuitable for detecting communities [17]. For instance, in a typical graph partitioning problem, the number of nodes to be partitioned and the number of groups to be partitioned are usually known in advance, which is of great difference from the community detection problem we are considering. A second serious drawback of these graph-partition-based methods is that all of them are essentially variations of finding partitions of a graph so that the number of crossing edges between partitions are minimized. However, small number of crossing edges alone may not be a good indication for communities without considering the intrinsic connections among the nodes in the graph.

Newman's revolutionary notion of *modularity* is the first successful attempt to resolve the drawbacks specified above [17]. The modularity is defined on a partition of the nodes in a graph. Let $G = (V, E)$ be an undirected graph modeling a social network with n nodes and m edges. Assume that each node v belongs to community c_v. We define the indicator function $\delta(c_u, c_v) = 1$ if and only if $c_u = c_v$, i.e., u, v are in the same community, and otherwise $\delta(c_u, c_v) = 0$ for two nodes $u, v \in V$. The modularity Q of this specific community partition is

$$Q = \frac{1}{2m} \sum_{u,v} \left(A_{uv} - \frac{d_u d_v}{2m} \right) \delta(c_u, c_v),$$

where A is the adjacency matrix of G with $A_{uv} = 1$ if $(u, v) \in E$ and 0 otherwise, and d_u, d_v are degrees of u and v respectively. The above definition follows Clauset et al.'s notation [4] and they also provided a detailed justification for defining modularity in this way. The modularity measurement is basically calculating the number of edges within the communities minus the expected number of such edges if we randomly rewire the same number of edges. Clauset et al. believe that optimizing the modularity corresponds to appropriately finding a family of communities. Although modularity based algorithms have been proven to perform well in many real world data, this family of approaches have several limitations. First, this approach implicitly assumes that communities do not intersect with one another, which is usually not the case for real world communities. Second, it is known that modularity based approaches suffer from the resolution limits [7], i.e., they are favorable to merge small communities.

Researchers also start to pay attention to the model-based methods for community detection. In this line of research, a generative model, depending on the underneath community structure, for the social network is assumed. For example, in Copie et al.'s work [5], they first assume that there is a fixed partition of the agents in the social network, where each partition is a community. Two agents within the same community is more likely to form an connection than those that are from different communities. When the social network is presented, one may infer the most likely partition through Bayesian law. One major question remains unsolved under this framework. Namely, the choice of generative models adds one more level of uncertainty since it is always difficult to validate the correctness of any specific model. Also, as we shall see, there is a major distinction between model-based methods like Copie et al.'s work [5] and ours. In model-based methods, the formation of the social network is usually the result of the agents' community structure while in our model, the agents choose to form community structure *after* the social network is formed.

Recently, a noticeable body of work is focusing on discovering *overlapping* community structures. In a pioneering work, Palla et al. [21] asserted that finding overlapping communities is highly relevant to finding k-cliques in the social networks. Their algorithm first finds all k-cliques in the network with a fixed constant k. Two k-cliques are connected if they share $k-1$ nodes, and each community is formed by a maximum set of connected k-cliques. Since one node may belong to multiple disconnected k-cliques, the final communities are overlapping with each other. Palla et al. showed that their algorithm has reasonable running time, despite the fact that computing cliques is hard. However, this method requires the size of clique as an input, which is usually hard

to provide. Furthermore, if the social network is highly connected, the algorithm will fail to uncover the underlying structure with any reasonable clique size. In fact, using k-clique to find overlapping communities implicitly assumes that no two communities share a $k-1$ clique.

Lancichinetti et al. [11] proposed to iteratively compute a local community from a node optimizing a *fitness* function, which is defined by the internal and external degrees of the computed subgraph. By varying the parameter in the fitness function, they obtain both overlapping and hierarchical community structures.

Another line of work in addressing the overlapping community is based on extensions of the modularity. Nocosia et al. [18] introduces a vector of belonging factors for each node in the graph, indicating the probability that this node belongs to a particular community. They then define a modularity measurement based on the belonging factors. One particular drawback of their algorithm is that it requires the number of communities as the input, which is not applicable in most cases. Furthermore, for each node, the total belonging factors summarize to 1, which essentially denies the idea that one node may just fit the communities as well as everyone else there.

The formation of communities was put in the game theoretic context by Athey and Jha [2], in which they explicitly addressed the losses and gains of associating oneself with a community. However, the social network underneath the agents in Athey and Jha's work is not in the picture. Their work is thus not comparable with ours.

The remaining of the paper is organized as follows. In Section 2, we formally define our game-theoretic framework in community detection. We propose an algorithm to simulate a local dynamic to reach local equilibria in Section 3. In Section 4, we conduct experiments on various benchmark and real world networks. We discuss the benefits and the limitation of our framework in Section 5.

2 A game-theoretic framework for community detection

As opposed to identifying the communities via attempting to globally optimize a certain measurement defined, we propose that we shall interpret the social network naturally, as a *community formation game* played by selfish agents. Every agent has her intrinsic utility that associates with which communities she joins and which she does not. The sheer goal every agent aims to achieve is to maximize her own utility. The formation of communities is thus the joint result of each agent's *selfish* decision. Interestingly, the formation of communities in the social network is never studied from the game-theoretic perspective.

The framework we are proposing is quite simple and natural. The social network is given, i.e., each agent's social interaction is fixed. There is an associated utility function defined on each agent for the set of communities she participates. We decompose the utility of the agent into two components. The first component is the gain, or pleasure, of an agent joining the communities. The second component is the loss, or pain, associated with the agent's action of joining the communities. It has been shown that joining a community provides one with tremendous benefits, physical or emotional [23]. Therefore, every agent shall have the incentive to join every single community unless there is an associated loss to joint the communities. And indeed, it is not hard to see in the real

world that joining a community usually associates with certain loss, e.g., membership fees. Finally, an agent's utility is merely the difference between her gain function and loss function.

2.1 Community formation game

In this section, we formally define our *community formation game*. We are given an underlying static acquaintance graph $G = (V, E)$, with $n = |V|$ and $m = |E|$. We assume G is undirected and unweighted though it is straightforward to generalize our results to directed and weighted graphs. Depending on the context, an element in V may either be called as an *agent* or a *node*. Each node chooses a collection of communities that it wants to join. The set of all possible communities is denoted as $[k] = \{1, 2, \ldots, k\}$, where k is polynomial in n (an exponential number of communities is both infeasible to detect in theory and unrealistic in practice). Notice that our final community structure may have much smaller number of communities than k. The utility of the ith agent $v_i \in V$ is measured by a gain function $g_i(\cdot)$ and a loss function $\ell_i(\cdot)$. Finally, for any family of functions $\{f_1, f_2, ..., f_n\}$ defined over the agents, let $f(\cdot) = \sum_{i\in[n]} f_i(\cdot)$, e.g., $g(\cdot) = \sum_{i\in[n]} g_i(\cdot)$ and $\ell(\cdot) = \sum_{i\in[n]} \ell_i(\cdot)$.

Strategy space and Nash equilibrium. In our community formation game, the strategies of agent v_i are subsets of communities that it wants to join, i.e, all subsets of $[k]$. We denote $L_i \subseteq [k]$ as a strategy of v_i, which we also refer to as the community label of v_i. We allow $L_i = \emptyset$, which means that i chooses to not belong to any community. Define $\mathcal{L} = (L_1, L_2, \ldots, L_n)$ as a strategy profile, which is a vector of community labels for all agents.

The utility of v_i is measured by a gain function $g_i(\cdot)$ and a loss function $\ell_i(\cdot)$, which map $\mathcal{L}$ to real numbers.[4] Let the community labels of agents other than i to be $\mathcal{L}_{-i}$, and we use $(\mathcal{L}_{-i}, L_i')$ to denote a strategy profile where the i-th entry of $\mathcal{L}$ is replaced by L_i'. We define the utility function for v_i to be: $u_i(\mathcal{L}) = g_i(\mathcal{L}) - \ell_i(\mathcal{L})$.

In the community formation game, given the strategies of other agents $\mathcal{L}_{-i}$, the *best response strategy (or strategies)* of agent v_i is:

$$\arg\max_{L_i'\subseteq[k]} g_i(\mathcal{L}_{-i}, L_i') - \ell_i(\mathcal{L}_{-i}, L_i').$$

Definition 1 (Pure Nash equilibrium). *Given graph G, the strategy profile* $\mathcal{L} = (L_1, L_2, \ldots, L_n)$ *forms a* (pure) Nash equilibrium *of the community formation game if all agents are playing their best strategies, that is,*

$$\forall i \text{ and } L_i' \neq L_i,\ u_i(\mathcal{L}_{-i}, L_i') \leq u_i(\mathcal{L}_{-i}, L_i).$$

In other words, in a Nash equilibrium, no agent can improve her own utility by changing her strategy unilaterally. One can interpret that each agent is satisfied with her community selection at the state of a Nash equilibrium. Since each node may select more than one community, the communities detected at the equilibrium naturally can be overlapping with each other, which shall reflect what occurs in the real world.

[4] A more appropriate notation should be $g_i^G(\cdot)$ and $\ell_i^G(\cdot)$. However, since the underlying graph G is static in our game, it is simpler to omit the superscript on G.

Existence and computation of Nash equilibria. In general, Nash equilibria may not exist in a community formation game. To see this, one can easily formulate a "matching pennies" game [20] in the community formation game, in which one node u always prefer to be with another node v in the same community while v always prefer not to be in the same community as u. It is thus interesting to know when a Nash equilibrium exists in the community formation game.

Let us recall that *potential games* are a general class of games that permit pure Nash equilibria [19]. In a potential game, there is an associated potential function $\Phi(\cdot)$ defined on the strategy profiles of the agents. A community formation game is a potential game if $\Phi(\mathcal{L}) - \Phi(\mathcal{L}_{-i}, L_i') = u_i(\mathcal{L}_{-i}, L_i') - u_i(\mathcal{L})$ for every strategy profile $\mathcal{L}$ and every strategy L_i' of v_i. In other words, if an agent changes her strategy to improve her own utility, the potential function strictly decreases with the same amount as the increase of the agent's utility. In any potential game that contains a finite number of strategy profiles, Nash equilibria always exist. Furthermore, every better response dynamic, in which each agent sequentially changes her strategy to improve her own utility, will converge to a Nash equilibrium.

Next, we provide a sufficient condition to make a community formation game potential, and thus address the existence of Nash equilibria for community detection purpose.

Definition 2. *A set of functions $\{f_i(\cdot) : 1 \leq i \leq n\}$ is* locally linear *with linear factor ρ if for every strategy profile $\mathcal{L}$ and every strategy L_i' of v_i, the following relation holds:*

$$\forall i \in [n],\ f_i(\mathcal{L}_{-i}, L_i') - f_i(\mathcal{L}) = \rho\left(f(\mathcal{L}_{-i}, L_i') - f(\mathcal{L})\right).$$

We show that if the gain and loss functions in a community formation game are *locally linear*, the game is a *potential game*.

Theorem 1. *Let $\{g_i(\cdot) : i \in [n]\}$ and $\{\ell_i(\cdot) : i \in [n]\}$) be the sets of gain and loss functions of a community formation game. If $\{g_i(\cdot)\}$ and $\{\ell_i(\cdot)\}$) are locally linear functions with linear factor ρ_g and ρ_ℓ, then the community formation game is a potential game.*

Proof. We define a potential function as $\Phi(\mathcal{L}) = \rho_\ell \cdot \ell(\mathcal{L}) - \rho_g \cdot g(\mathcal{L})$. Now consider agent v_i who changes her strategy from L_i to L_i'. From the definitions of *locally linear functions* and the utility functions $u_i(\cdot)$, we have $\Phi(\mathcal{L}) - \Phi(\mathcal{L}_{-i}, L_i') = u_i(\mathcal{L}_{-i}, L_i') - u_i(\mathcal{L})$. Therefore, the community formation game is a potential game. □

With locally linear gain and loss functions, we can guarantee the existence of Nash equilibria, but it is not necessarily true that finding a Nash equilibrium is easy. The follow lemma indicates that computing a Nash equilibrium could be hard. Due to space limitation, we omit the proof of the lemma.

Lemma 1. *There exists a community formation game, in which the sets of gain and loss functions are locally linear, such that both computing the best response for an individual agent and computing a Nash Equilibrium in the game are NP-hard.*

Gain and loss functions. We now propose a set of gain and loss functions. These gain and loss functions have natural economic interpretations and they can be computed efficiently. Additionally, our experiments also demonstrate that the equilibria using these gain and loss functions provide pretty accurate information regarding the community formation.

The gain function we use here is a generalized version of the well accepted modularity function, which is adapted to fit the scenario that one node may participate in multiple communities. We define $\hat{\delta}(i,j) = 1$ if $|L_i \cap L_j| \geq 1$ and $\hat{\delta}(i,j) = 0$ otherwise. Let A be the adjacency matrix of graph G.

Definition 3 (Personalized modularity function). *The personalized modularity function defined for the i-th agent is:*

$$Q_i(\mathcal{L}) = \frac{1}{2m} \sum_{j \in [n]} \left(A_{ij} \hat{\delta}(i,j) - \frac{d_i d_j}{2m} \cdot |L_i \cap L_j| \right).$$

Similar to the original modularity, the personalized modularity function measures the number of edges from this agent to the community comparing with such number of edges if all edges within the community are randomly sampled according to the degree constrains on the agents. In particular, it describes how well this agent fits the community comparing with a random community. Note that the original modularity can be obtained by summing over the personal modularity functions for all agents, under the original non-overlapping condition of $|L_i| = 1$ for all $i \in [n]$. We use $|L_i \cap L_j|$ instead of $\hat{\delta}(i,j)$ in the second term so that we have $\sum_{j \in [n]} d_j \cdot |L_i \cap L_j| = \sum_{c \in L_i} \sum_{j \in [n], c \in L_j} d_j$, which allows fast local computation in our algorithm (see Section 3).

We use a simple loss function to model the aspect that an agent may suffer by joining new communities. This, in particular, reflects some fixed cost associated in joining a new community. In real world, it may connect to the membership fee or some cost to attend community events.

Definition 4 (Linear loss function). *Let $c > 0$ be a constant. The loss of a node v_i with the linear loss function is $(|L_i| - 1) \cdot c$.*

It is easy to verify that both the personalized modularity function and the linear loss function are *locally linear functions*, with linear factor $1/2$ and 1, respectively. Therefore, we have the following result.

Theorem 2. *Let $g_i(\mathcal{L})$ be the personalized modularity function and $\ell_i(\mathcal{L})$ be a linear loss function. The community formation game has a Nash equilibrium.*

3 Local equilibrium and a simple algorithm

We have shown that if the set of gain and loss functions are *locally linear*, there always exists a Nash equilibrium in our community formation game (Theorem 1). However, computing the best response might be hard in some simple cases from Lemma 1. It is thus not reasonable to assume that individuals always make the best response. Instead,

Algorithm 1 LocalEquilibrium(G)

1: initialize each node to a singleton community
2: repeat the following process until no node can improve itself
3: randomly pick a node v_i, and perform the best operation among *join, leave* and *switch*

we propose that an agent will only choose a strategy from a restricted space that depends on her current state when she needs to respond to the other agents' strategies. In particular, an agent can only locally implement the following three operations,

1. *join.* Agent v_i joins a new community on top of the communities she joins by adding a new label in L_i.
2. *leave.* Agent v_i leaves a community she is in by removing a label from L_i.
3. *switch.* Agent v_i switches from one community to another by replacing a label in L_i.

In the restricted strategy spaces, an equilibrium is a state where no agent can deviate from her current strategy within the locally allowed strategy space. Such kind of equilibria are referred as *local equilibria* [1] in the literature. In the community formation game, the entire strategy space of agent i is $\mathcal{S} = 2^{[k]}$. For each agent i with the current community label set L_i, we use $ls(L_i)$ to denote i's local strategy space, which is the set of possible label sets we could obtain by applying one of the operations *join, leave* and *switch* once on L_i. The local equilibrium is defined as follows:

Definition 5 (Local equilibrium). [5] *Given G, the strategy profile $\mathcal{L} = (L_1, L_2, \ldots, L_n)$ forms a* local equilibrium *of the community formation game if all agents are playing their local optimal strategies, that is,*

$$\forall i \text{ and } L_i' \in ls(L_i),\ u_i(\mathcal{L}_{-i}, L_i') \leq u_i(\mathcal{L}_{-i}, L_i).$$

Different from Nash equilibrium, at local equilibrium, the utility of each agent is achieving a *local maximum* instead of a global one. This is useful when the local strategy space is easy to explore, while computing a global optimal solution is not feasible. Restricting strategy space to our local strategy space is further justified by the fact that in real world individuals usually consider joining or leaving one community at a time.

In our setting, computing the local best response in the local strategy space is polynomial-time, by simply enumerating all $O(k)$ possible *join, leave* and *switch* operations. We show in the following lemma that for the case of using our personalized modularity gain function and linear loss function, the computation can be made efficient by maintaining a quantity for each community, and by only checking new communities to add or switch to from the set of communities of one's neighbors.

Lemma 2. *Let Δ_i be the degree of the node v_i and N_i be the set of neighbors of v_i in the graph G. Let $g_i(\mathcal{L})$ be the personalized modularity function and $\ell_i(\mathcal{L})$ be a linear loss function. The time complexity to find the best local operation for agent i in* join, leave *and* switch *is $O(|L_i| \cdot |L(N_i)| \cdot \Delta_i)$, where $L(N_i) = \cup_{j \in N_i} L_i$.*

[5] The *local equilibrium* in [1] is defined on Euclidean strategy spaces.

Proof. It is sufficient to show that the function $Q_i(\mathcal{L})$ can be efficiently updated. For each community U, we maintain a quantity $\hat{Q}_U = \sum_{v_j \in U} \frac{d_j}{2m}$. Notice that if only one member leaves U or joins U in each step, $\hat{Q}_U$ can be updated in constant time.

For the *join* operation, if v_i is joining a community not in $L(N_i)$, $Q_i(\mathcal{L})$ as well as $u_i(\mathcal{L})$ will be strictly decreasing. Therefore, v_i only considers the communities in $L(N_i)$ to *join*. For the *switch* operation, the same argument applies, except that v_i may have a possible gain if she switches to a brand new community U', but in this case v_i can gain the same utility by simply removing the old community. Therefore v_i only needs to consider communities in $L(N_i)$.

In joining a new community, $Q_i(\mathcal{L})$ can be updated in $O(\Delta_i)$ time. This is because the term $-\sum_{j \in U} \frac{d_i d_j}{2m}$ can be computed in constant time given the maintained $\hat{Q}_U$. The *switch* operation will consider leaving $|L_i|$ communities and joining $|L(N_i)|$ communities. Therefore, the total running time is $O(|L_i| \cdot |L(N_i)| \cdot \Delta_i)$. □

The fast computation of the local best operation is important in our experiment. This is in particular a reason we propose to use $|L_i \cap L_j|$ in defining the personalized modularity function. Next, we use the simple algorithm LocalEquilibrium(G) (Algorithm 1) to compute a local equilibrium, which essentially simulates the best local response dynamic. Although in general the local response dynamic may take long time to converge, we show below that for the case of the personalized modularity function and the linear loss function, the convergence is fast if we set the parameter c in the linear loss function appropriately.

Theorem 3. *Let $g_i(\mathcal{L})$ be the personalized modularity function and $\ell_i(\mathcal{L}) = c(|L_i|-1)$ be a linear loss function with constant c satisfying $4cm^2$ is an integer.* LocalEquilibrium *takes at most $O(m^2)$ steps to reach a local equilibrium.*

Proof. Define the potential function $\Phi(\mathcal{L}) = \ell(\mathcal{L}) - g(\mathcal{L})/2$, where $\ell(\mathcal{L}) = \sum_i \ell_i(\mathcal{L})$ and $g(\mathcal{L}) = \sum_i g_i(\mathcal{L})$. Notice that $\Phi(\mathcal{L}) \geq -1/2$ since $g(\mathcal{L}) \leq 1$ and $\ell(\mathcal{L}) \geq 0$.

Now consider $4m^2 \cdot \Phi(\mathcal{L})$. It is straightforward to verify that $4m^2 \cdot \Phi(\mathcal{L})$ is an integer function. Notice that we have $\Phi(\mathcal{L}) = 0$ in the initial state of LocalEquilibrium, and $\Phi(\mathcal{L}) \geq -1/2$ during the algorithm execution. Since $4m^2 \cdot \Phi(\mathcal{L})$ strictly decreases in each round, $\Phi(\mathcal{L})$ will achieve minimum in at most $2m^2$ steps. □

Our LocalEquilibrium algorithm uses the initial configuration in which every agent has one community of her own. One reason we choose this starting point instead of other possible ones, such as all nodes sharing one single community, is that with this starting point most of the agents' activities will be joining communities, which is likely to be an individual decision. In contrast, when people share one community and the community evolves by splitting, the splitting decision is usually a collective decision made by a group of people, which is not modeled in our current game-theoretic framework and it would be computationally expensive to incorporate group decisions. Our experimental results show that this starting point indeed leads to the extraction of reasonable community structures. Additionally, if we already have some partial knowledge on who are in the same communities, we can choose it as our starting point and do not allow the local dynamic to change it, and thus community learning can be naturally incorporated in our framework as well. This is a future direction we will pursue next.

4 Experiments

It is very hard to obtain ground truth community information from real-world networks. Therefore, detecting communities sometimes is more of an art than a science [5]. Nevertheless, we conduct experiments on some real-world networks as well as several benchmark graphs. In the experiment, our algorithm uses the personalized modularity functions as the gain functions and the simple loss functions with the loss factor $c = \frac{1}{m}$.

4.1 Real world graphs

We run our algorithm on two classical graphs: *The Dolphin Network* [12] and *The Zachary's Karate Club* [25]. We present our decomposition of the communities in Figure 1. In particular, we believe that our algorithm finds richer overlapping structures compared with previous studies [17, 18]. For example, Newman uses modularity to partition the same karate club network into two components [17], which corresponds to two upper overlapping communities and the four lower communities we discovered in Figure 1 (b). Thus, our community structure is a strict refinement of the community structure discovered in Newman's work. Our refined structure allows overlapping communities, which by visual inspection does provide meaningful information about the community structures.

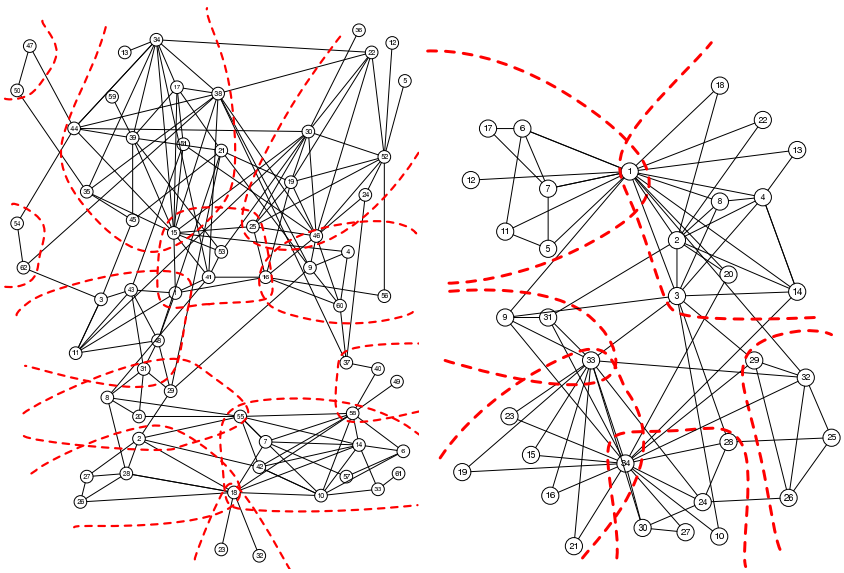
For some small communities detected by our algorithm (e.g. nodes 47 and 50 in the upper-left corner of Figure 1 (a)) could be merged with their neighbor communities if we consider joint decisions of all community members. Considering joint decisions is more computational intensitive, however, and we defer it as a future research item.

Our conclusion from this experiment is that our algorithm is able to discover finer grained communities than previous approaches. These finer grained communities reveals insightful community structures in the network and could be used in further investigation of community interconnections and larger community structures.

4.2 Benchmark graphs

Recently, Lancichinetti and Fortunato [11] proposed a set of benchmark graphs to evaluate the performance of the community detection algorithms that can discover overlapping communities. The evaluation metric used is the "normalized mutual information" between the recovered community structure and the underlying ground truth. Though it could be rather hard to justify how realistic these benchmark graphs and the evaluation metric are, we show that our method actually performs pretty well in the benchmark graphs.

In this data set, we compare our algorithm with the clique percolation method [21] and the CONGA algorithm [8]. The clique percolation algorithm requires the size of the cliques as input. We run the algorithm with different clique sizes ranging from 2 to 6. The results presented in Figure 2 are obtained by taking the optimal value for the clique sizes. The CONGA algorithm on the other hand, requires the number of communities as input. Although this information is available here, in general, we do not know the number of communities. Another disadvantage of CONGA is its slow running time. This is due to the computation of the "betweenness" values at each node,



(a) (b)

Fig. 1. The communities discovered for the dolphin network (left). The communities discovered for the Zachary's karate club (right).

which essentially has to compute all pairs of shortest paths. In particular, we are only able to obtain results of CONGA for the graphs with 1,000 nodes.

In Figure 2, we show the results of different algorithms on the benchmark graphs with overlapping communities. The networks used to produce the left figure consist of 1,000 nodes, whereas those of the right figure consist of 5,000 nodes. In each figure, the community sizes of the two upper diagrams range between $s_{min} = 10$ and $s_{max} = 50$, and the community sizes of two bottom diagrams range between $s_{min} = 20$ and $s_{max} = 100$. The mixing parameter, i.e., the portion of crossing edges, μ is 0.1 for two left diagrams and 0.3 for two right diagrams. The other parameters are $\tau_1 = 2$, $\tau_2 = 1$, $k_{avg} = 20$, $k_{max} = 50$ and $om = 2$. The x-axis represents the portion of nodes that belong to multiple communities. The graphs are generated in the following steps: (1) generate the number of communities that each node will belong to; assign the degrees to the nodes based on a power law distribution with exponent τ_1; (2) assign the community sizes from another power law distribution with exponent τ_2 for a fixed number of communities; (3) generate the bipartite graph between the nodes and the communities with the configuration model [14]; (4) for each node, assign the cross-community degree and internal degrees within each community based on μ; (5) build the graphs for each community and the cross-community edges with the configuration model. For more detailed steps on how the graphs are generated and refined, readers can refer to [11].

As shown in Figure 2, our algorithm for the game-theoretic framework performs better for the bottom two networks. In other words, our algorithm works better on

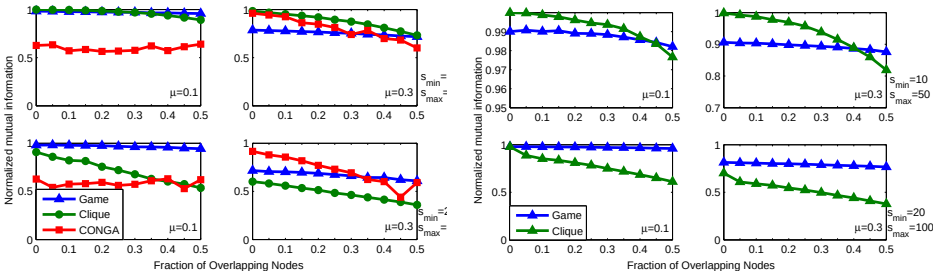


Fig. 2. The network on the left consists of 1,000 nodes. The minimum degree and maximum degree of the network are 10 and 50 respectively. The network on the right consists of 5,000 nodes. The minimum degree and maximum degree of the network are 10 and 50 respectively.

graphs with larger communities. Notice that the community sizes in the bottom two graphs are within 20 and 100, while the upper two graphs have communities with size 10 to 50.

Compared with the clique percolation algorithm, both our algorithm and the clique percolation algorithm perform very well on the two upper left networks, with mutual information being above 90%. For the two upper right networks in Figure 2, the clique percolation algorithm outperforms our algorithm when the fraction of overlapping nodes is small. However, our algorithm is more stable than the clique percolation algorithm over all instances. The performance of the clique percolation algorithm drops significantly when the portion of overlapping nodes increases. In particular, when half nodes belong to multiple communities (at the point 0.5 on the x-axis), the performance of our algorithm actually is equally good to the clique percolation algorithm for graphs with 1,000 nodes, and performs better on graphs with 5,000 nodes.

Compared with CONGA algorithm, our algorithm is better for $\mu = 0.1$, and is not as good as CONGA for $\mu = 0.3$. Again the performance of our algorithm is more stable than CONGA, and is better when the fraction of overlapping nodes is large. Also notice that CONGA is not able to finish in reasonable time for graphs with 5000 nodes.

4.3 Identifying duplicated names in DBLP

Finally, we provide an application scenario of the game-theoretic based community detection algorithms, which suggests our community-detection approach may extend well beyond the notion of "communities" defined in traditional ways.

The study of co-authorship network in academic community has attracted much attention recently [15, 16]. Examples of co-authorship databases include Microsoft Academic Search and the DBLP computer science bibliography. Extracting the co-authorship graph from the existing databases like DBLP sometimes could be challenging. For example, different scholars with the same name may be naively viewed as a single person. While people using Indo-European languages do not often experience this problem, people from countries like China, Japan, or Korea, whose names in Microsoft Academic or DBLP are indeed Romanization of their original names, usually find the problem of having duplicated names persistent.

Therefore, it is interesting to find a way to distinguish different people with the same name in academic community when relevant data, say, records from DBLP, are presented. Let us now present one possible way that serves as the first step to tackle this problem using the game-theoretic based algorithms we developed. First, we construct a co-authorship graph based on DBLP entries. A node in the graph corresponds to a name, and one node may represent more than one person in the real world. Two nodes are linked by an undirected edge if the corresponding names of these two nodes ever coauthored at least one paper. Next, in this co-authorship graph, each node is asked to play the community-formation game using the personalized modularity gain functions and linear cost functions until a local equilibrium is reached. One would naturally expect that even when people with the same name collapse into one node, the node will join multiple communities in the game because these people with the same name shall belong to different communities. Based on this observation, a node may be partitioned into multiple entities and an individual represented by this node may correspond to one or more entities.

We provide one instance of experiment which demonstrates our observation is reasonably accurate. Our experiment searches for the node with name "Wei Chen" in the co-authorship graph, which in fact represents more than 20 individuals that have published in total more than 200 papers in computer science or relevant areas. We use only a subgraph of the co-authorship graph that contains 20,000 nodes because processing the whole graph would otherwise be too computationally intensive. The subgraph is obtained by using breadth first search from the node "Wei Chen" until 20,000 nodes are discovered. The data is for publications until the end of year 2008.We specifically focus on two "Wei Chen"s: one is the first author of this paper from Microsoft Research Asia (MSRA) and the other is a faculty member in Zhejiang University (ZJU).

Our algorithm discovers more than 40 communities containing "Wei Chen". In most communities, we find that the set of co-authors are relevant to a particular "Wei Chen". For example, we find four communities that are relevant to the two "Wei Chen"s we mentioned above. Table 1 summarizes the four sets of co-authors of two "Wei Chen"s in the four communities.

Wei Chen (MSRA)	Jialin Zhang, Chao Jin, Zheng Zhang, Likun Liu, Shiding Lin, Ming Chen, Shaomei Wu, Yu Chen, Qiao Lian, Ben Y. Zhao, Xuezheng Liu
	Marcos Kawazoe Aguilera, Sam Toueg
Wei Chen (ZJU)	William M. Andrews, Aidong Lu, David S. Ebert, Mario Costa Sousa, Ross Maciejewski, Tobias Isenberg
	Zhongding Jiang, Yi Gong, Yu Guan, Jin Wang, Yingchao Zhao, Chunxiao Liu, Zi'ang Ding, Guofeng Zhang, Yingzhen Yang, Ling Zhuang, Hongxin Zhang, Chengfang Song, Huafeng Liu, Huagen Wan, Luying Li, Hujun Bao, Xiao Liang, Qunsheng Peng, Qifeng Tan, Pengcheng Shi, Yubo Zhang, Shang-Hua Teng, Lincan Zou, Xiaobo An,Xueying Qin, Long Zhang, Yinan Fan, Dong Xu, Yun Zeng, Wei Hua, Zhao Dong,

Table 1. The partition of the co-authors of two "Wei Chen"s

The first "Wei Chen" is the first author of this paper. The co-authors of him are splitted into two communities. One related to his research collaborators after he joins Microsoft Research Asia, and the other are his collaborators back when he was at Cornell.

The second "Wei Chen" is a faculty member in Zhejiang University. The first group of his collaborators represents his connection in Purdue university. The second group of the co-authors is his colleagues in Zhejiang University, with the exception of "Shang-Hua Teng" and "Yingchao Zhao". These two authors are actually the co-authors of "Wei Chen (MSRA)". A reason to explain the misclassification is that Teng and Zhao only co-authored one paper with "Wei Chen (MSRA)" (by the end of 2008) and they did not collaborate with any other "Wei Chen (MSRA)"s co-authors. On the other hand, Teng had collaboration with "Harry Shum" that has strong connections with the graphics researchers, one of whom is "Wei Chen (ZJU)". In this respect, it is actually hard to say that it was a "misclassification" since the two authors only collaborate with "Wei Chen (MSRA)" once.

5 Conclusion

We propose for the first time a game-theoretic framework to detect community structures in social networks. This formulation intuitively matches the dynamic formation of communities in real world scenarios. Furthermore, since we do not require each agent to join exactly one community, the resulting community structure natually incorporates overlapping communities.

Our experiment shows that, even with simple ulitity functions defined on the agents, our method is effective in discovering overlapping communities in several benchmark graphs and real world networks. Since the algorithm we use to find local equilibrium only implements local operations, the running time is fast and the algorithm can fit into a parallel framework.

There remain many interesting open problems under this framework. One direction is to find more appropriate gain and loss functions. The proposed ones in this paper, though simple and effective, are by no means the best choices for the community formation games. In particular, we believe better gain and loss functions can be obtained by deeper understanding of the community formation process in the real world networks.

6 Acknowledgement

The authors thank Prof. Wei Chen from State Key Lab of CAD&CG, Zhejiang University for commenting on the partition produced by our algorithm.

References

1. C. Alós-Ferrer and A. Ania. Local equilibria in economic games. *Economics Letters*, 70(2):165–173, 2001.
2. S. Athey and S. Jha. A theory of community formation and social hierarchy. working paper, 2006.

3. U. Brandes and T. Erlebach. *Network Analysis: methodological foundations*. Springer Verlag, 2005.
4. A. Clauset, M. E. J. Newman, and C. Moore. Finding Community Structure in Very Large Networks. *Phys. Rev. E*, 70(6):066111, Dec 2004.
5. J. Copic, M. O. Jackson, and A. Kirman. Identifying Community Structures from Network Data via Maximum Likelihood Methods. *The B.E. Journal of Theoretical Economics*, 9, 2009. working paper.
6. P. O. Fjällström. Algorithms for graph partitioning: A Survey. In *Linkoping Electronic Atricles in Computer and Information Science, 3.*, 1998.
7. S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36, 2007.
8. S. Gregory. A fast algorithm to find overlapping communities in networks. In *ECML/PKDD*. Springer, 2008.
9. J. D. Kasarda and M. Janowitz. Community Attachment in Mass Society. *American Sociological Review*, 39(3):328–339, 1974.
10. P. Kotler and G. Zaltman. Social Marketing: An Approach to Planned Social Change. *The Journal of Marketing*, 35(3):3–12, 1971.
11. A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):16118, 2009.
12. D. Lusseau. The emergent properties of a dolphin social network. *Proceedings: Biological Sciences*, 270:S186–S188, 2003.
13. D. McKenzie-Mohr and W. Smith. *Fostering Sustainable Behavior: An Introduction to Community-Based Social Marketing*. New Society Publishers, 1999.
14. M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6(2-3):161–180, 1995.
15. M. E. J. Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5200–5205, 2004.
16. M. E. J. Newman. Who Is the Best Connected Scientist?A Study of Scientific Coauthorship Networks. *Complex Networks*, 650:337–370, 2004.
17. M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.
18. V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech*, 3024, 2009.
19. N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.
20. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
21. G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814, 2005.
22. R. J. Sampson and W. B. Groves. Community Structure and Crime: Testing Social-Disorganization Theory. *American Journal of Sociology*, 94(4):774, 1989.
23. S. Sarason. *The Psychological Sense of Community*. Jossey-Bass, 1974.
24. H. C. White, S. A. Boorman, and R. L. Breiger. Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions. *American Journal of Sociology*, 81(4):730, 1976.
25. W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.