Community Detection in Social Networks through Community Formation Games*

Wei Chen [†] and **Zhenming Liu** [‡] and **Xiaorui Sun** [§] and **Yajun Wang** [†]

Abstract

We introduce a game-theoretic framework to address the community detection problem based on the social networks' structure. The dynamics of community formation is framed as a strategic game called *community formation game*: Given a social network, each node is selfish and 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. Each agent can select multiple communities, which naturally captures the concept of "overlapping communities". We propose a gain function based on Newman's modularity function and a simple loss function that reflects the intrinsic costs incurred when people join the communities. We conduct extensive experiments under this framework; our results show that our algorithm is effective in identifying overlapping communities, and is often better than other algorithms we evaluated especially when many people belong to multiple communities.

1 Introduction

Understanding the formation and evolution of communities is a long-standing research topic in sociology in part because of its fundamental connections with the studies of urban development, criminology, social marketing, and several other areas [8; 17; 12]. With increasing popularity of online social network services like Facebook, the study of community structures assumes more significance. Identifying and detecting communities are not only of particular importance but have immediate applications. For instance, for effective online marketing, such as placing online ads or deploying viral marketing strategies [10], identifying communities in social network could often lead to more accurate targeting and better marketing results. Albeit online user profiles or other semantic information is helpful to discover user segments, this kind of information is often at a coarse-grained level and overlooks community structure that reveals rich information at fine-grained level.

We study community detections based on the network structure. There exists extensive research work on detecting communities of this kind, for which we provide a more detailed introduction in the next subsection. From these researches, however, we find a couple of issues that we feel are unsatisfactory. First, many community detection algorithms either set a global optimization goal for detection, such as optimizing for *modularity*, *betweenness*, or *conductance* [13; 7; 4], or only look for sub-components with certain predetermined structures [16]. 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 centralized authority enforcing a global objective. Second, a majority of the researches focus on partitioning graphs into disjoint components and thus cannot accommodate overlapping communities. In social networks, an individual typically belongs to more than one communities, such as the community of her family members, that of her friends, and that of her coworkers. A satisfactory community detection algorithm shall incorporate overlapping communities in a natural way.

We address the above two issues by borrowing game-theoretic concepts often used in economics researches. We model each node in a social network as an autonomous and rational agent. Based on the network structure, joining and leaving communities will result in certain benefits and costs to an agent. The agents' only incentive is to optimize their net payoff by joining appropriate communities. We coin this model as *community formation game* (CFG). 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 strategy.

In general, computing the best response in a CFG could be intractable. To address this issue, we adopt the notion of *local equilibra* [2], i.e. local optimal, as our solution concepts. In a local equilibrium, intuitively, no agent is able to increase its payoff without radically changing its strategy. Using local

*Original version of this paper appeared in Data Mining and Knowledge Discovery, 21(2), 2010.

[†]Microsoft Research, weic, yajunw@microsoft.com.

[‡]Harvard, zliu@harvard.edu.

[§]Shanghai Jiaotong University, sunsirius@gmail.com.

equilibria serves dual purposes. First, it is more plausible that agents exhibit bounded rationality in real world settings; second, efficient algorithms can usually be designed to find local equilibrum. We believe our game theoretic framework reflects real-world organic community formations, and thus in principle suggests a more systematic approach than existing ones. This framework also naturally incorporates overlapping communities because we allow an individual to join multiple communities as long as this could improve her payoffs.

In our framework, an agent's utility 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 a community detection algorithm, we address a number of issues, in particular the existence and the computational efficiency of Nash equilibria in a CFG.

We show that in general Nash equilibria may not exist but we provide a sufficient condition to permit the existence of it. Specifically, we define a family of functions called locally linear functions and show that if all gain and loss functions of the agents are locally linear, the game is guaranteed to have a Nash equilibrium. For computational efficiency, we show that some locally linear gain and loss functions will result in games for which even finding the best response is intractable. To bypass this problem, we focus on finding local equilibria in a CFG. We propose a natural set of gain and loss functions that are both locally linear and allow efficient computation of a local equilibria. Our gain function is based on a "personalized" version of modularity [13], and thus reflects the individual desire of forming a close-knit community structure. Our loss function is based on the number of communities one need to maintain.

We conduct experiments on both synthetic and real networks to evaluate our algorithm. We adopt synthetic benchmark graphs in [9] to evaluate our algorithm and compare it with two other recent algorithms that allow overlapping communities [16; 7]. The results demonstrate that our algorithm performs quite well overall, and is usually better than others when a considerable portion of nodes belong to multiple communities. As a proof of concept application, our algorithm is also used to disambiguate Chinese authors with the same name in DBLP and our results are promising.

1.1 Related work

Early work in graph partitions could be adopted as algorithms for community detection. However, Newman pointed out a few facts that make the approaches unsuitable in general [13]. For instance, these algorithms usually require the knowledge on the number of partitions as part of the input, which is unrealistic in community detection problems. Another drawback is that these algorithms usually find partitions to minimize cross edges. Small number of crossing edges alone may not be a good indication for communities without considering the intrinsic connections among the nodes in a graph.

Newman's notion of *modularity* is the first successful attempt to resolve the drawbacks specified above [13]. The modularity is defined on a partition of the nodes in a graph. Intuitively, the modularity is calculating the number of edges within the communities minus the expected number of such edges if we randomly rewire the same number of edges. In spite of its good performance on many real world data, this family of approaches has several limitations: it assumes that communities do not intersect with one another. Also, modularity based approaches usually have the "resolution limit" problems [6], i.e., they favor larger communites.

Generative models (e.g. [5]) are also developed to solve the community detection problem. Maximum-likelihood methods are used to find the underlying communities. We feel that the choice of generative models adds one more level of uncertainty since validating models are difficult. Also, there is a major distinction between these model-based methods and ours. In model-based methods, the formation of the social network is usually decided by the community structure while in our model, the agents choose to form communities *based on* the structure of a social network.

Recently, overlapping community problems have also been investigated. Palla et al. [16] reduces this problem to finding k-cliques. This method requires to know the size of cliques, which is usually unrealistic. Furthermore, if the social network is highly connected, the algorithm will fail to uncover the underlying structure with any reasonable clique size. Lancichinetti et al. [9] proposed to iteratively compute a local community from a node to optimize a *fitness* function, which is defined by the internal and external degrees of the computed subgraph. By varying the parameters in the fitness function, they obtain both overlapping and hierarchical community structures. Nocosia et al. [14] extends the notion of modularity. They introduce 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. The algorithm also requires to know the number of communities, which we think is a drawback.

The formation of communities was put in a game theoretic context by Athey et al. [3], in which they explicitly addressed the losses and gains of associating oneself with a community. However, the social network's topology is not in the picture. Their work is thus not comparable with ours. Adjeroh et al. [1] used game theoretic analysis as a way to initialize the community structure. Their concept of community is defined on the energy landscape theory.

2 The game-theoretic framework

We propose that the formation of communities shall be interpreted as a *community formation game* played by selfish agents on the social network. Each agent has her intrinsic utility that associates with the communities she joins and those she does not. Individuals only aim to maximize their own utility. The formation of communities is the joint result of each agent's *selfish* decision.

We now elaborate the framework. First, we assume the link structure of a social network is available, i.e., each agent's social interaction is known and fixed. Each agent is associated with a utility function which depends on the set of communities she decides to join. The utility of an agent consists of

two components: the *gain* (or pleasure) of joining the communities and the *loss* (or pain) to do so. Joining a community usually provides one with tremendous benefits, physical or emotional [18]. Therefore, each individual shall have the incentive to join all the communities unless there is some cost to do so. Indeed, in the real world, certain loss or pain usually incurs when one joins a community. For instance, a person might find it time consuming to maintain the connections or friendships with other members in the same community; the cost can also be in monetary form, e.g., membership fee required to retain her status in an academic society.

2.1 Community formation game

We now define the *community formation game*. Let $G = (V, E)$ be a social network, where $|V| = n$ and $|E| = m$. We assume G is undirected and unweighted though generalization of our results to directed or weighted graphs is possible. An element in V is sometimes also called as an *agent*. Each agent chooses a collection of communities that it wants to join. The set of all possible communities is denoted as $[k] = \{1, 2, ..., k\}$, where k is polynomial in n. We remark that an exponential number of communities is both infeasible to detect in theory and unrealistic in practice. It is possible that no agent decides to join a certain community and this community becomes empty. Therefore, our final community structure may have much smaller number of communities.

Strategy space and Nash equilibrium. In a CFG, the strategies of agent v_i are subsets of communities that it joins, i.e., $2^{[k]}$. We denote $L_i \subseteq [k]$ as a strategy of v_i, which we also refer as the community label of v_i. We allow $L_i = \emptyset$, which means that v_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.[1] Let the community labels of agents other than i be $\mathcal{L}_{-i}$; let $(\mathcal{L}_{-i}, L_i')$ be a strategy profile, where the i-th entry of $\mathcal{L}$ is replaced by L_i'. Define the v_i's utility as $u_i(\mathcal{L}) = g_i(\mathcal{L}) - \ell_i(\mathcal{L})$.

The gain function and loss function depend on both the network structure and the community structure. However, they shall not depend on the labeling of the communities. i.e., any relabeling of the communities (or permutation on $[k]$) shall not result in a change of the gain and loss functions' values. For any family of functions $\{f_1, f_2, ..., f_n\}$ (with the same domain) defined over the agents, let $f(\cdot) = \sum_{i \le n} f_i(\cdot)$. Specifically, $g(\cdot) = \sum_{i \le n} g_i(\cdot)$ and $\ell(\cdot) = \sum_{i \le n} \ell_i(\cdot)$.

In a CFG, 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 2.1 (Pure Nash equilibrium). *Given a 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,* $(\forall i \text{ and } L_i' \neq L_i,\ u_i(\mathcal{L}_{-i}, L_i') \le u_i(\mathcal{L}_{-i}, L_i))$.

[1]The utility functions shall also depend on G but we surpass this dependency because G is always clear from the context.

In a Nash equilibrium, no agent can improve her own utility by changing her strategy unilaterally. We 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.

Existence and computation of Nash equilibria. In general Nash equilibria may not exist in a CFG. To see this, one can easily formulate a "matching pennies" game [15] as a CFG, 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 a CFG. Let us recall that *potential games* are a general class of games that permit pure Nash equilibria [15]. In a potential game, there is an associated potential function $\Phi(\cdot)$ defined on the strategy profiles. A CFG 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. i.e.,when an agent changes her strategy to improve her utility, the potential function strictly decreases with the same amount as the increase of the agent's utility. In any potential game with a finite number strategy profiles, Nash equilibria always exist. Furthermore, if in a potential game each agent sequentially changes her strategy to improve her utility, the strategy profile will converge to a Nash equilibrium.

We now provide a sufficient condition to make a CFG potential, and thus address the existence of Nash equilibria for community detection purpose.

Definition 2.2. *A set of functions $\{f_i(\cdot) : 1 \le i \le n\}$ is* locally linear *with linear factor ρ if for every strategy profile $\mathcal{L}$ and every strategy L_i' of v_i, we have* $(\forall i \in [n],\ f_i(\mathcal{L}_{-i}, L_i') - f_i(\mathcal{L}) = \rho\left(f(\mathcal{L}_{-i}, L_i') - f(\mathcal{L})\right))$.

A concrete example of the locally linear functions will be given in Definition 2.5.

Theorem 2.3. *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.*

The potential is set as $\Phi(\mathcal{L}) = \rho_\ell \cdot \ell(\mathcal{L}) - \rho_g \cdot g(\mathcal{L})$ to prove the theorem. Though Nash equilibria always exists when gain and loss functions are locally linear, finding the best response could still be hard in this case.

Lemma 2.4. *There exists a CFG, in which gain and loss functions are all locally linear, such that computing the best response for an individual is 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. Our experiments also demonstrate that the equilibria using these gain and loss functions provide useful information regarding the community formation.

We generalize the well accepted modularity function as our gain function. Define $\hat{\delta}(i, j) = 1$ if $|L_i \cap L_j| \ge 1$ and $\hat{\delta}(i, j) = 0$ otherwise. Let A be the adjacency matrix of G.

Definition 2.5 (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 an agent to the community comparing with such number of edges if all edges within the community are randomly sampled subject to the degree constraints on the agents. Observe that Newman's modularity can be described as the sum of every agent's personal modularity functions under a non-overlapping condition, i.e., $|L_i| = 1$ for all $i \le n$. Substituting the $|L_i \cap L_j|$ term by $\hat{\delta}(i,j)$ is possible but the former one allows us to compute the personalized modularity more efficiently.

We use a simple loss function to model the aspect that an agent may suffer by joining new communities. This reflects some fixed cost associated in joining a new community.

Definition 2.6. *Let $c > 0$ be a constant. The loss of a node v_i with the linear loss function (LLF) is* $(|L_i| - 1)\cdot c$.

Both the personalized modularity and the linear loss function are locally linear (with linear factors being $\frac{1}{2}$ and 1).

Theorem 2.7. *Let $g_i(\mathcal{L})$ be the personalized modularity function and $\ell_i(\mathcal{L})$ be a LLF. The CFG has a Nash equilibrium.*

We remark that any type of loss functions that only depend on $|L_i|$ are locally linear. Another interesting loss function could be a concave function in $|L_i|$, which exhibits diminishing marginal cost property as one joins more communities.

3 Local equilibrium and a simple algorithm

It is unreasonable to assume individuals always make the best response because computing it could be intractable even when the gain and loss functions are locally linear. 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. Specifically, an agent can only locally implement the following three operations,

1. *join.* An agent v_i joins a new community on top of the communities she joins by adding a new label in L_i.
2. *leave.* An agent v_i leaves a community she is in by removing a label from L_i.
3. *switch.* An 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* [2] in the literature. In the CFG, 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.

Definition 3.1 (Local equilibrium). *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,* $(\forall i \text{ and } L_i' \in ls(L_i),\ u_i(\mathcal{L}_{-i}, L_i') \le u_i(\mathcal{L}_{-i}, L_i))$.

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*

The Local equilibrium is useful when the local strategy space is easy to explore while computing a global optimal solution is not feasible. Algorithm 1 illustrates a way to find a local equilibrium efficiently.

Theorem 3.2. *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.*

Our LocalEquilibrium algorithm uses the initial configuration in which every agent has one community of her own. One reason we choose this starting point 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 framework. If there is 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.

4 Experiments

In the experiment, our algorithm uses the personalized modularity function as the gain functions and the simple loss function with the loss factor $c = \frac{1}{m}$.

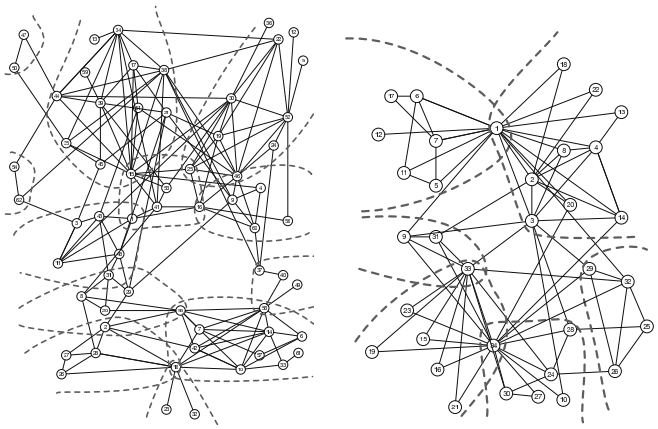
4.1 Real world graphs

We run our algorithm on two graphs: *The Dolphin Network* [11] and *The Zachary's Karate Club* [19]. Figure 1 illustrates the results of our algorithm. We feel that our algorithm finds richer overlapping structures compared with previous studies [13; 14]. For example, Newman uses modularity to partition the same karate club network into two components [13], 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 [13].

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. Enumerating joint decisions is more computational intensive; understanding it is left as a future research item.

4.2 Benchmark graphs

We adopt the benchmark graphs proposed in [9] to evaluate our algorithm's performance for the overlapping community detection. The evaluation metric is the "normalized mutual information" between the recovered community structure and the underlying ground truth.



(a) (b)

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

We compare our algorithm with the clique percolation method and the CONGA algorithm [16; 7]. The clique percolation algorithm requires the size of the cliques as input. We tried different clique sizes ranging from 2 to 6 and we only present the optimal value. The CONGA algorithm, on the other hand, requires the number of communities as input. Another disadvantage of CONGA is its slow running time due to the computation of "betweenness" for each node, which is not able to work with 5000 nodes.

The experimental results are shown in Figure 2. The networks for the upper (resp. lower) subfigure consist of 1,000 (resp. 5000) nodes. In each subfigure, the community sizes of the two upper (resp. lower) diagrams range between $s_{min} = 10$ (resp. 20) and $s_{max} = 50$ (resp. 100). The mixing parameter, i.e., the portion of crossing edges, μ is 0.1 (resp. 0.3) for two left (resp. 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. See [9] for more details on how these graphs are generated based on the parameters.

Our algorithm for the game-theoretic framework performs better for the bottom two networks. This suggests our algorithm could work better on graphs with larger communities. Compared with the clique percolation algorithm, both ours and theirs perform 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 ours when x is small. However, our algorithm is more stable than the clique percolation one over all instances. The performance of the clique percolation algorithm drops considerably when x increases. When half nodes belong to multiple communities (at the point 0.5 on the x-axis), the performance of our algorithm becomes equally good to the clique percolation one 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 worse than CONGA for $\mu = 0.3$. Again the performance of our algorithm is more stable than CONGA, and is better when x is large.

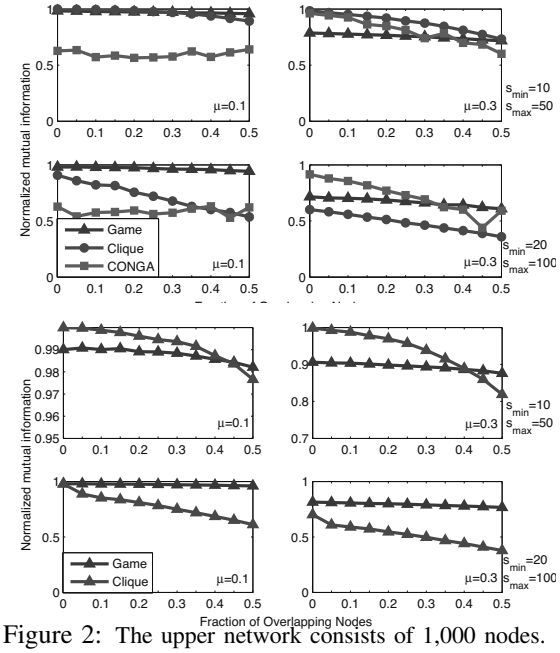


Figure 2: The upper network consists of 1,000 nodes. The minimum degree and maximum degree of the network are 10 and 50 respectively. The lower network consists of 5,000 nodes. The minimum degree and maximum degree of the network are 10 and 50.

4.3 Identifying duplicated names in DBLP

Finally, we provide an application scenario of the game-theoretic based community detection algorithm, which suggests our community-detection approach may extend well beyond the notion of "communities" defined in traditional ways. As our goal is to demonstrate a proof-of-concept application, we do not attempt to compare results here with other relevant works.

Consider the problem of extracting co-authorship network from data sets like DBLP. A challenge of this problem is different scholars may share the same name/identity in DBLP data and we need to disambiguate the names to produce an accurate co-authorship network. We adopt our algorithm to solve this problem in the following way. 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 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 in real world.

We provide one instance of experiment which demonstrates our algorithm is promising. 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

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,

Figure 3: The partition of the co-authors of two "Wei Chen"s

science or relevant areas. We use a subgraph of the co-authorship graph that contains 20,000 nodes by breadth first search from the node "Wei Chen". 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). Table 3 summarizes the four sets of co-authors of two "Wei Chen"s in the four communities by our algorithm.

The first "Wei Chen" is the first author of this paper. The co-authors of him are split into two communities. One relates 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", which 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)" (as of 2008). On the other hand, Teng had collaboration with "Harry Shum" that has strong connections with "Wei Chen (ZJU)".

5 Conclusion

We proposed 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 naturally incorporates overlapping communities. Our experiment shows that simple utility functions already permit us to effectively discover overlapping communities. 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. Better gain and loss functions can be derived by deeper understanding of the community formation process in real world.

References

[1] D. Adjeroh and U. Kandaswamy. Game-Theoretic analysis of network community structure. *International Journal of Computational Intelligence Research*, 3(4):313–325, 2007.

[2] C. Alós-Ferrer and A. Ania. Local equilibria in economic games. *Economics Letters*, 70(2):165–173, 2001.

[3] S. Athey and S. Jha. A theory of community formation and social hierarchy. working paper, 2006.

[4] U. Brandes and T. Erlebach. *Network Analysis: methodological foundations*. Springer Verlag, 2005.

[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] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36, 2007.

[7] S. Gregory. A fast algorithm to find overlapping communities in networks. In *ECML/PKDD*. Springer, 2008.

[8] J. D. Kasarda and M. Janowitz. Community Attachment in Mass Society. *American Sociological Review*, 39(3):328–339, 1974.

[9] 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.

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

[11] D. Lusseau. The emergent properties of a dolphin social network. *Proceedings: Biological Sciences*, 270:S186–S188, 2003.

[12] D. McKenzie-Mohr and W. Smith. *Fostering Sustainable Behavior: An Introduction to Community-Based Social Marketing*. New Society Publishers, 1999.

[13] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[14] 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.

[15] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[16] 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.

[17] R. J. Sampson and W. B. Groves. Community Structure and Crime: Testing Social-Disorganization Theory. *American Journal of Sociology*, 94(4):774, 1989.

[18] S. Sarason. *The Psychological Sense of Community*. Jossey-Bass, 1974.

[19] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.