

Community Detection in Dynamic Social Networks: A Game-Theoretic Approach

Hamidreza Alviri
Department of EECS
University of Central Florida
halvari@eecs.ucf.edu

Alireza Hajibagheri
Department of EECS
University of Central Florida
alireza@eecs.ucf.edu

Gita Sukthankar
Department of EECS
University of Central Florida
gitars@eecs.ucf.edu

Abstract—Most real-world social networks are inherently dynamic and composed of communities that are constantly changing in membership. As a result, recent years have witnessed increased attention toward the challenging problem of detecting evolving communities. This paper presents a game-theoretic approach for community detection in dynamic social networks in which each node is treated as a rational agent who periodically chooses from a set of predefined actions in order to maximize its utility function. The community structure of a snapshot emerges after the game reaches Nash equilibrium; the partitions and agent information are then transferred to the next snapshot. An evaluation of our method on two real world dynamic datasets (AS-Internet Routers Graph and AS-Oregon Graph) demonstrates that we are able to report more stable and accurate communities over time compared to the benchmark methods.

Keywords—community detection; dynamic social networks; game-theoretic models

I. INTRODUCTION

The natural flux of people’s changing social ties and interests expressed on online social networking sites generates a dynamic social network. As the network changes, user communities evolve and can grow, shrink, or disappear. Here, the word *community* refers to the emergent groups that are created as people form highly connected networks with their families, co-workers, and friends. In this context, we regard the communities as simply friendship networks in which the participants share common interests, activities, or a geographic location. Intuitively we expect more edges inside the community compared to its outside, i.e., intra-connections tend to be more common than inter-connections.

Community detection can help us understand the hidden social structure of the user populations, but the dynamic aspect of networks can pose problems for standard algorithms. This paper leverages game-theoretic models of rational behavior to attack the problem of dynamic community detection. The proposed approach for solving the problem treats each node within the observed snapshots of a given network as a selfish agent who tries to maximize its utility (the difference of its personal gain and loss functions) in a game-theoretic way. The Nash equilibrium of the current game corresponds to the final community structure of the current snapshot. All of the communities discovered in previous snapshots are then propagated to the current time

slice and used along with the network structure of the current snapshot. The main contributions of this paper are twofold:

- We introduce the Dynamic Game Theory method (**D-GT**), which extends the game-theoretic model described in [1] to the problem of community detection in dynamic social networks. D-GT treats the nodes of the network as rational agents which perform a set of actions in a iterative and game-theoretic way to maximize their total utility.
- We validate our method on two dynamic datasets, the AS-Internet Routers Graph and AS-Oregon Graph, and compare the results with those of two other methods, LabelRankT and iLCD, evaluating the modularity metric.

In the remainder of the paper, first the related work is presented in Section II. In Section III we delve into the problem and explain the proposed method. The experiments are discussed in Section IV, and finally we conclude the paper in Section V.

II. RELATED WORK

The problem of community detection has appeared in various forms across several disciplines including physics, sociology, and computer science. Many efficient methods have been proposed for detecting both non-overlapping and overlapping communities on static networks, among which CFinder [2] and COPRA [3] have distinguished themselves as among the most popular and effective methods, assuming they are initialized with good parameters [4]. Approaches range from traditional network structure based algorithms [5]–[7] to more modern optimization algorithms employing game theory [1], label propagation [8]–[10], and information diffusion [11], [12]. Detecting community structure in dynamic networks, on the other hand, has attracted less research attention due to the complexity of the problem and dearth of good datasets.

Similar to the static community detection approaches, methods that have been proposed to detect communities in dynamic networks can be categorized into two main groups: traditional and modern methods. Traditional algorithms use general information about the network properties to perform community detection. Hui et al. [13] proposed a distributed method for community detection in which modularity was

used as a measure instead of the objective function. [14] attempted to track the evolution of communities over time, using a few static network snapshots. Palla et al. [15] proposed an innovative method for detecting communities in dynamic networks based on the k-clique percolation technique. This approach can detect overlapping communities; however, it is time consuming, especially on large-scale networks. iLCD [16] can identify overlapping network communities by adding edges and then merging similar ones. However, this model cannot always handle dynamic behaviors of the network, particularly when new nodes are introduced or removed, or when existing edges are removed.

In addition to the traditional algorithms, there are some approaches that use techniques such as cost optimization, label propagation, random walk, and information theory to solve the problem of dynamic detection of community structures. FacetNet proposed in [17] is a framework for analyzing communities in dynamic networks based on optimizing snapshot costs. FacetNet is guaranteed to converge to a locally optimal solution; however, its convergence speed is slow, and it needs to be initialized with the number of communities, which is usually unknown in practice. Recently, Xie et al. [18] introduced a stabilized label propagation algorithm named LabelRankT. LabelRankT is an extension of LabelRank in which each node requires only local information during label propagation processing. In this paper, we benchmark our algorithm against both iLCD and LabelRankT.

The Markov Cluster Algorithm (MCL) proposed in [19] uses simulations of flow (random walk) and repeatedly executes matrix multiplications followed by an inflation operator. Regularized-MCL [20] is an extension of MCL that uses a regularization operator to avoid overfitting the data with too many communities. The authors observed that this approach still suffers from the scalability issue of the original MCL. In order to overcome this problem, they introduced a multi-level version of regularized MCL in which the graph is successively coarsened prior to running regularized MCL. Jimeng et al. [21] presented a parameter-free methodology for detecting clusters on time-evolving graphs based on mutual information and entropy.

In contrast, our method, D-GT, attempts to simulate the decision-making process of the individuals creating the communities, rather than focusing on statistical correlations between labels of neighboring nodes. We believe that exploiting game theory for dynamic community detection yields more realistic, fine-grained communities since intrinsically game theory is a good representation for expressing the behavior of individuals and strategic interactions among them [22]. In previous work, we have demonstrated the success of game-theoretic approaches in static community detection across several domains, including detecting guilds in massively multiplayer online games [23] and predicting trust between users on e-commerce sites [24].

Table I
DEFINITION OF SYMBOLS

Symbol	Definition
T	Snapshots
G_t	A graph with no self-edges at snapshot T_t
m_t, n_t	Number of edges and vertices at snapshot T_t
A_t	Adjacency matrix at snapshot T_t
S	Profile of strategies
s_i	Strategy of agent i
g_i	Gain function of agent i
l_i	Loss function of agent i
u_i	Utility function of agent i
C_{ij}	Similarity between agents i and j

Table II
DEFINITION OF POSSIBLE ACTIONS.

Action	Definition
<i>Join</i>	Add a new label to s_i
<i>Leave</i>	Remove a label from s_i
<i>Switch</i>	Remove a label from s_i and add a new one
<i>No Action</i>	No specific action is performed

III. METHOD

In this section, we present the framework for D-GT; our source code is freely available at <https://github.com/hamidalvari/D-GT/>. Table I shows the symbols and definitions used throughout the paper.

Suppose that we have M snapshots $T = \{T_0, T_1, \dots, T_M\}$ of the given network with their corresponding underlying graphs $G_t = (V_t, E_t)$, with $n_t = |V_t|$ vertices and $m_t = |E_t|$ edges where $t \in \{1, 2, \dots, M\}$. We extend the work described in [1] by considering the process of community detection as an iterative game performed in a dynamic multi-agent environment in which each node of the underlying graph is a selfish agent who decides to maximize its total utility u_i . In other words, we use an agent-based model to detect communities by optimizing each user's utility through a stochastic search process.

During the game, each agent can periodically take an action (*join*, *switch*, *leave*, or *no operation*) to change its community membership. Specifically, it decides whether to join a new community C by adding its label to s_i which results in the utility gain u_{Join} according to (1):

$$s_i \leftarrow s_i \cup \{C\}. \quad (1)$$

Another option is to leave one of its own communities, e.g., C' , by removing its label from s_i which results in the utility gain u_{Leave} :

$$s_i \leftarrow s_i / \{C'\}. \quad (2)$$

The agent can also decide to switch communities, say C' , by deleting its label from s_i , adding the new community, C , and receiving u_{Switch} as its utility:

$$s_i \leftarrow s_i / \{C'\}, s_i \leftarrow s_i \cup \{C\}. \quad (3)$$

Finally, the new utility u'_i for this agent is calculated and the previous utility u_i is replaced by the new one:

$$u'_i \leftarrow \max\{u_{Join}, u_{Leave}, u_{Switch}, u_{noOp}\}. \quad (4)$$

The set of all communities is denoted by $[k] = 1, 2, \dots, n$. We define a strategy profile $S = (s_1, s_2, \dots, s_n)$ which represents the set of all strategies of all agents, where $s_i \subseteq [k]$ denotes the strategy of agent i , i.e. the set of its labels.

In our framework, for each snapshot, the best response strategy of an agent i with respect to strategies S_{-i} of other agents is calculated as: $\arg \max_{s_i \subseteq [k]} u_i(S_{-i}, s_i)$. The utility function for each agent is calculated by:

$$u_i(S_{-i}, s_i) = g_i(S_{-i}, s_i) - l_i(S_{-i}, s_i), \quad (5)$$

where the gain function for agent i is formulated as follows:

$$g_i(S) = \frac{1}{m_t} \sum_{l=1}^{|s_i|} \sum_{j \in L, j \neq i} C_{ij}. \quad (6)$$

For calculating the similarities between each pair of vertices in G , we can use local or global properties, regardless of whether or not the nodes are directly connected. In this paper, we use neighborhood similarity [1] to quantify the structural similarity between users:

$$C_{ij} = \begin{cases} w_{ij}(1 - d_i d_j / 2m_t) & A_{ij,t} = 1, w_{ij} \geq 1 \\ w_{ij}/n_t & A_{ij,t} = 0, w_{ij} \geq 1 \\ d_i d_j / 4m_t & A_{ij,t} = 1, w_{ij} = 0 \\ -d_i d_j / 4m_t & A_{ij,t} = 0, w_{ij} = 0 \end{cases} \quad (7)$$

where w_{ij} is the number of common neighbors shared by node i and j and d_i is the degree of node i . C_{ij} assumes its highest value when two nodes have at least one common neighbor and are also directly connected, i.e. $A_{ij,t} = 1$.

Similar to what happens in real life, we also consider the loss function l_i for each agent, which is linear in the number of labels each agent has, since community membership incurs implicit overhead costs (e.g., fees) for each individual. Therefore we define the following loss function for agent i :

$$l_i(S_{-i}, s_i) = \frac{1}{m_t} (|s_i| - 1). \quad (8)$$

The strategy profile S forms a pure Nash equilibrium of the community formation game if all agents play their best strategies. Algorithm 1 shows our proposed framework. After calculating similarities between each pair of agents (Equation 7), the multi-agent game commences. The community structure of each snapshot of the network emerges after agents reach the local equilibrium. For each snapshot T_k , we use existing agents and labels from all of its previous snapshots $T_{[1..k-1]}$ along with new agents corresponding to the new nodes that were just added to the current snapshot, if any. Propagating community labels from all the

previous snapshots provides the algorithm with a large set of initial candidate communities and improves the convergence speed. In the version of the algorithm for static community detection, each agent is initialized as the lone member of a single community and must discover new communities through the use of join or switch operations.

Algorithm 1 D-GT: Dynamic Game Theoretic Model

```

1: Input: Snapshots  $T = \{T_0, T_1, \dots, T_M\}$  of the network
2: Output: Communities of each snapshot
3: for all snapshots do
4:    $t \leftarrow \text{current\_snapshot}$ 
5:   Construct the list agents of graph  $G_t$ 
6:   Calculate similarities  $C_{ij}$  between pairs of vertices of graph  $G_t$ 
7:   while NOT convergence in the agents' utilities do
8:      $\text{agent}_i = \text{Random\_Select}(\text{agents})$ 
9:      $\text{action}_i = \text{Best\_Operation}(\text{join}, \text{leave}, \text{switch})$ 
10:     $u'_i = \text{Utility\_Calculate}(\text{agent}_i, \text{action}_i)$ 
11:    if  $u_i < u'_i$  then
12:       $u_i \leftarrow u'_i$ 
13:      Update  $s_i$ 
14:      Update communities
15:    else
16:       $\text{action}_i = \text{noOp}$ 
17:    end if
18:  end while
19:  Send the labels to the snapshot  $T_{t+1}$ 
20: end for

```

IV. EXPERIMENTS

All of the algorithms were evaluated together on a system with 12G of RAM and Intel CPU 2.53 GHz for the purpose of fair comparison. The following state-of-the-art community detection approaches are used as benchmarks:

- **LabelRankT**¹ [18]. LabelRankT functions according to the generalized LabelRank, in which each node requires only local information during label propagation processing. Several parameters must be set before running the algorithm on the data; we used the best performing values reported by the authors [18].
- **iLCD**² [16]. iLCD is a well known community detection approach for dynamic social networks which works by first adding edges and then merging the similar ones. It takes the dynamics of the network into account when grouping nodes into communities.

A. Datasets

To illustrate the strength and effectiveness of our approach, we selected two popular real-world social networks

¹<https://sites.google.com/site/communitydetectionslpa>

²http://www.cazabetremy.fr/Cazabet_remy/iLCD.html

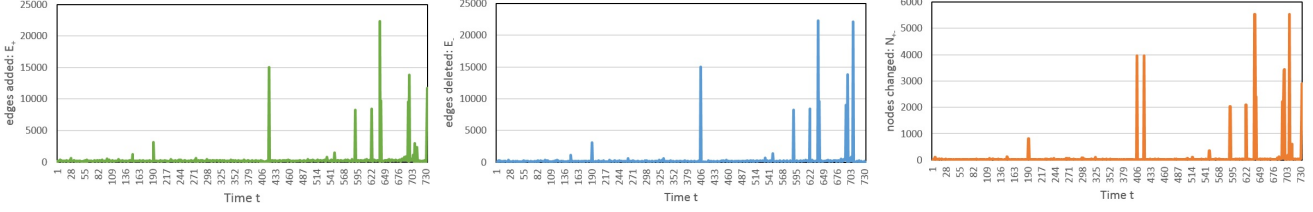


Figure 1. The structural changes over 733 snapshots in the AS-Internet dataset, including the number of edges added (E_+ , left image) and deleted (E_- , middle image), as well as the number of nodes involved in changes (N_{+-} , right image).

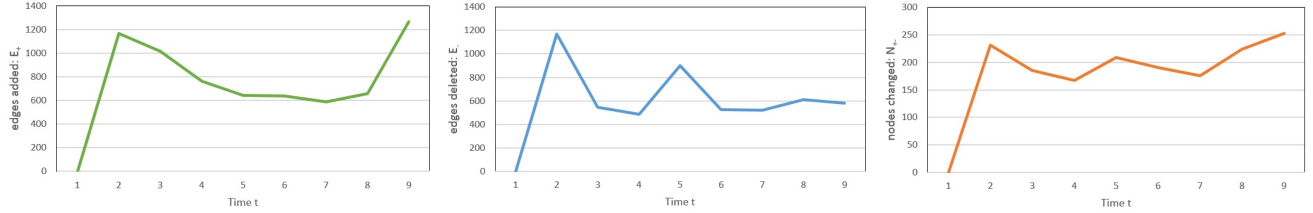


Figure 2. The structural changes in the AS-Oregon dataset over 9 snapshots, including the number of edges added (E_+ , left image) and deleted (E_- , middle image), as well as the number of nodes involved in changes (N_{+-} , right image).

from the SNAP³ graph library. AS-Internet and AS-Oregon were chosen from the available datasets, since they have a variable number of snapshots. The large number of nodes in both datasets demonstrates that our approach is capable of running on large datasets. The description of the datasets is as follows:

AS-Internet Routers Graph [25]. This is a communication network of who-talks-to-whom from the Border Gateway Protocol logs of routers in the Internet. The dataset contains 733 daily snapshots collected between November 8, 1997 to January 2, 2000. The number of nodes in the largest snapshot is 6,477 (with 13,233 edges). In contrast to citation networks, where nodes and edges only get added (not deleted), the AS dataset exhibits both the addition and deletion of nodes and edges over time. The structure of the graph can change dramatically at each snapshot (Fig. 1). This figure illustrates the number of edges added and deleted, as well as the number of nodes involved in these changes.

AS-Oregon Graph. The dataset contains 9 graphs of Autonomous Systems (AS) peering information inferred from Oregon route-views between March 31, 2001 and May 26, 2001. These 9 graphs are different snapshots from the data with a minimum of 10,670 and maximum of 11,174 nodes. In addition, the number of edges ranges from 21,999 in the snapshot of April 07, 2001 to 23,409 in May 26, 2001. Figure 2 shows the number of edges added and deleted, as well as the number of nodes involved in the changes for the AS-Oregon dataset.

Table III
DATASET SUMMARY

Data	AS-Internet Routers Graph	Oregon
min # of nodes	2,948	11,174
max # of nodes	6,477	10,670
min # of edges	3,386	21,999
max # of edges	13,233	23,409
# of snapshots	733	9
edge type	directed	directed

B. Evaluation

Since the ground truth of neither of the two datasets is available, it is impossible to use normalized mutual information (NMI) to calculate the performance of the different algorithms. Hence, for measuring the performance, we use modularity [26] to evaluate our algorithm on the above mentioned datasets. Modularity is the most popular qualitative measure in detecting communities in social networks. However, it has been shown that modularity has drawbacks and becomes unreliable when our networks are too sparse [27]. Modularity Q is defined as follows:

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta_{c_i, c_j} \quad (9)$$

where A_{ij} is an element of the adjacency matrix, δ_{ij} is the Kronecker delta symbol, and c_i is the label of the community to which vertex i is assigned. However, modularity is slightly different for directed networks [28] and can be reformulated as:

³<http://snap.stanford.edu/>

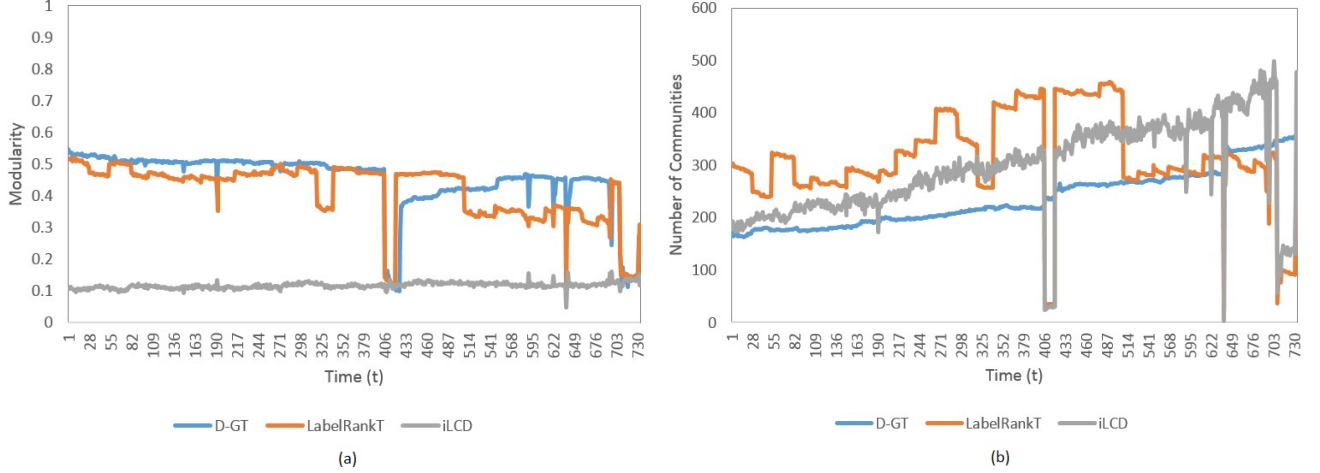


Figure 3. (a) D-GT modularity on the AS-Internet dataset compared with LabelRankT and iLCD. Dataset consists of 733 different snapshots. (b) Number of communities generated by D-GT algorithm in comparison with LabelRankT and iLCD.

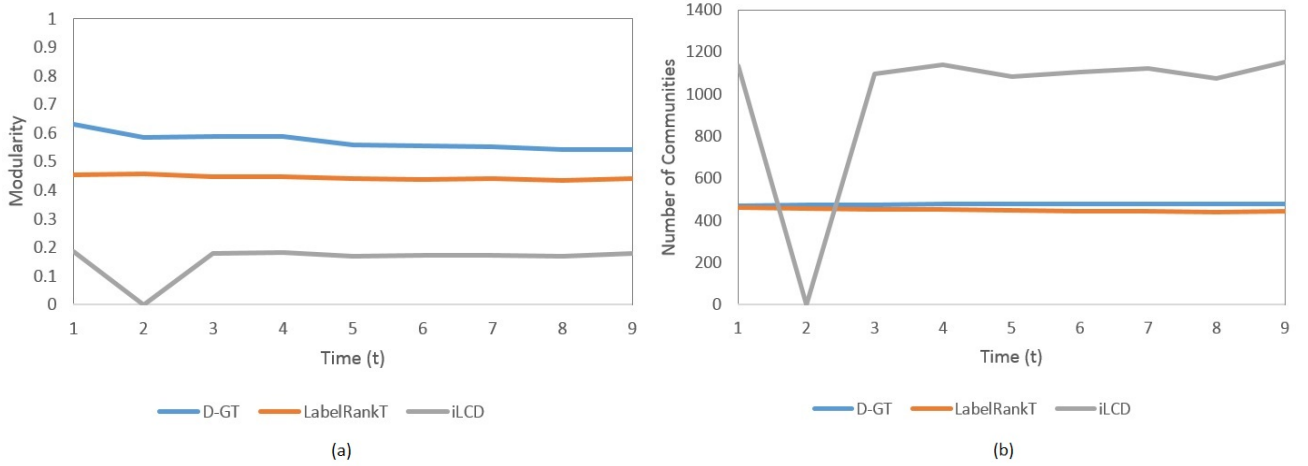


Figure 4. (a) D-GT modularity on the AS-Oregon dataset compared with LabelRankT and iLCD. Dataset consists of 9 different snapshots. (b) Number of communities generated by D-GT algorithm in comparison with LabelRankT and iLCD.

$$Q = \frac{1}{m} \sum_{ij} [A_{ij} - \frac{k_i^{in} k_j^{out}}{m}] \delta_{c_i, c_j} \quad (10)$$

where A_{ij} is defined in the conventional manner to be one if there is an edge from i to j and zero otherwise. Here, the probability of the existence of an edge from vertex i to vertex j is $k_i^{in} k_j^{out} / m$, where k_i^{in} and k_j^{out} are the in- and out-degrees of the vertices respectively.

Beside the modularity, the number of discovered communities is also used to compare the results of different algorithms. The number of communities, if any, within the network is typically unknown, however, an algorithm which discovers many more communities in a network is known to be more accurate. This can be explained by the fact that some methods disregard small communities and tend

to combine them in order to maximize the overall network modularity.

C. Results

We first report the modularity values corresponding to the D-GT, LabelRankT, and iLCD algorithms and then analyze the number of communities detected by these approaches. Results demonstrate that D-GT performs well on both datasets in comparison with the other two methods.

1) *AS-Internet*: Fig. 3 shows the modularity of the proposed method compared to the other two algorithms on the AS-Internet dataset. As can be easily determined, D-GT achieves competitively higher modularities than LabelRankT in most of the snapshots, while iLCD fails to find strong community structure at all. In particular, there is almost a continuous increase of the modularity values obtained by D-GT, since our method continues preserving

the community structure of the previous snapshots and only considers current network changes. Retaining the basic network community structure is a significant advantage of D-GT: it avoids the overhead of recomputing from scratch and hence runs faster. Moreover, D-GT delivers more consistent results in terms of number of communities as opposed to the LabelRankT and iLCD which both show dramatic fluctuations in the number of communities over time due to the changes in the structure of the underlying network.

2) *AS-Oregon*: AS-Oregon has many fewer snapshots compared to the AS-Internet, however, the number of nodes and edges are high enough for an extensive analysis. We compare modularity results obtained by D-GT at each network snapshot with those of LabelRankT and iLCD. Fig. 4 reveals that the modularities returned by D-GT are significantly higher than the results of LabelRankT and are still much more stable than those of iLCD. In addition to that, our method outputs a finer network community structure since it discovers more communities than LabelRankT.

On the other hand, the number of discovered communities by iLCD is higher than both other approaches and here is an explanation for that: as shown in Fig. 4, there is a linear correlation between modularities delivered by iLCD and the number of communities on this dataset. This suggests that iLCD generates more communities at the expense of modularity, creating highly disconnected communities that may not correspond to real groups.

V. CONCLUSION AND FUTURE WORK

This paper presents a game-theoretic approach to the problem of detecting communities in dynamic social networks. Our results demonstrate that the proposed method not only outperforms two other well known methods in terms of modularity but also in identifying evolving and fine-grained network communities. In future work, we plan to replicate the study with other utility functions, loss functions, and update rules; for instance, Q-learning could be used as the update rule for the agents.

Moreover, we are seeking to replace our similarity measure with one based on social features. Deploying such a measure might give more accurate and realistic results, since in the real world, individuals with similar social activities tend to form communities more frequently. Networks from some data sources, such as Massively Multiplayer Online Games (MMOGs), can contain a higher proportion of isolated nodes, and it would be beneficial to leverage other feature sources to handle these disconnected nodes.

ACKNOWLEDGMENTS

Research at University of Central Florida was supported by NSF award IIS-0845159.

REFERENCES

- [1] H. Alvari, S. Hashemi, and A. Hamzeh, "Detecting overlapping communities in social networks by game theory and structural equivalence concept," in *Artificial Intelligence and Computational Intelligence*. Springer Berlin Heidelberg, 2011, pp. 620–630.
- [2] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [3] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.
- [4] L. Peel, "Estimating network parameters for selecting community detection algorithms," in *Conference on Information Fusion (FUSION)*. IEEE, 2010, pp. 1–8.
- [5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [6] M. E. Newman, "Modularity and community structure in networks," *Proc Natl Acad Sci USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.
- [7] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, pp. 066133+, 2004.
- [8] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, 2007.
- [9] J. Xie and B. K. Szymanski, "Community detection using a neighborhood strength driven label propagation algorithm," in *Network Science Workshop (NSW)*. IEEE, 2011, pp. 188–195.
- [10] J. Xie and B. Szymanski, "Towards linear time overlapping community detection in social networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, LNAI, 2012.
- [11] A. Hajibagheri, A. Hamzeh, and G. Sukthankar, "Modeling information diffusion and community membership using stochastic optimization," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, pp. 175–182.
- [12] A. Hajibagheri, H. Alvari, A. Hamzeh, and S. Hashemi, "Community detection in social networks using information diffusion," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2012, pp. 702–703.
- [13] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of the ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, 2007, p. 7.

- [14] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *Proceedings of the National Academy of Sciences of the USA*, vol. 101, no. Suppl 1, pp. 5249–5253, 2004.
- [15] G. Palla, P. Pollner, A.-L. Barabási, and T. Vicsek, "Social group dynamics in networks," in *Adaptive Networks*. Springer Berlin Heidelberg, 2009, pp. 11–38.
- [16] R. Cazabet, F. Amblard, and C. Hanachi, "Detection of overlapping communities in dynamical social networks," in *International Conference on Social Computing (SocialCom)*. IEEE, 2010, pp. 309–314.
- [17] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," in *Proceeding of the International Conference on World Wide Web*. ACM, 2008, pp. 685–694.
- [18] J. Xie, M. Chen, and B. K. Szymanski, "LabelrankT: Incremental community detection in dynamic networks via label propagation," *arXiv preprint arXiv:1305.2006*, 2013.
- [19] S. Van Dongen, "A cluster algorithm for graphs," *Report-Information systems*, no. 10, pp. 1–40, 2000.
- [20] V. Satuluri and S. Parthasarathy, "Scalable graph clustering using stochastic flows: applications to community discovery," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 737–746.
- [21] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "Graphscope: parameter-free mining of large time-evolving graphs," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 687–696.
- [22] D. Adjeroh and U. Kandaswamy, "Game-theoretic analysis of network community structure," *International Journal of Computational Intelligence Research*, vol. 3, no. 4, 2007.
- [23] H. Alvari, K. Lakkaraju, G. Sukthankar, and J. Whetzel, "Predicting guild membership in massively multiplayer online games," in *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, Washington, D.C., April 2014, pp. 215–222.
- [24] G. Beigi, H. Alvari, M. Jalili, and G. Sukthankar, "Leveraging community detection for accurate trust prediction," in *ASE International Conference on Social Computing*, Palo Alto, CA, May 2014.
- [25] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 177–187.
- [26] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review*, vol. E 69, no. 026113, 2004.
- [27] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.
- [28] E. A. Leicht and M. E. Newman, "Community structure in directed networks," *Physical Review Letters*, vol. 100, no. 11, p. 118703, 2008.