COMMUNITY STRUCTURE AND INFORMATION PROPAGATION IN A TWITTER NETWORK

Yihan Guan[1], Yiliang Jin[2]

December 8, 2010

ABSTRACT

In this project, we analyze the community structure and the pattern of information propagation in a Twitter network with 60K nodes and 1.5M edges. We compare the performances of three modularity-based community detection algorithms and develop a modified Blondel-Guillaume-Lambiotte-Lefebvre's (BGLL) algorithm that can improve the computation efficiency of BGLL. The resulting communities detected by the modified BGLL algorithm are verified by modularity and conductance. Besides, we investigate the propagation patterns of different types of topics spreading in a Twitter network. Specifically, we analyze the information-spreading of sudden topics with different initial influencers within a community as well as in an entire network. The results demonstrate that not only different types of starting nodes can affect the propagation progress, but different types of topics can also influence the pattern of propagation.

1.INTRODUCTION

The modern science of networks has brought significant advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure, i.e., the organization of nodes in communities with many edges joining nodes of the same community and comparatively few edges joining nodes of different communities [2]. Detecting communities is of great importance in a broad range of disciplines where systems are often represented as graphs, such as physics, sociology, biology, and computer science. Therefore, in this project, we look for how to detect different social groups in a large Twitter network and reveal its structure features. Three algorithms are applied: Clauset-Newman-Moore's algorithm (CNM) [6], the BGLL algorithm [1], and a modified version of BGLL that we designed to improve the computation efficiency.

Meanwhile, blogs have become a significant media of communication and information propagation on the internet. Information spreads quickly and widely through blogs due to their broad accessibility and timely nature [4]. Twitter is a recent popular social networking and micro blogging service which provides a large-scale network for users to spread their own ideas or follow ideas that they like. In particular, different initial influencers in such a social network could reach final influenced groups of different sizes. To explore this phenomenon, in this project we investigate how information in a Twitter network spreads, how to identify the most influential nodes and how to choose the best initial propagation set to spread different types of information fast and widely.

[1] Department of Management Science and Engineering. Email: yihan@stanford.edu.
[2] Department of Computer Science. Email: yljin@stanford.edu

The objective of this project is two-fold. The first phase of this project aims at finding the most efficient algorithm to detect communities in a large Twitter network. Subsequently, the second phase of the project targets at investigating information propagation within the network, looking for the most influential nodes, and exploring patterns of propagation of various types of news in a community as well as in an entire network.

2. Data and Algorithms

2.1 Dataset Description

Twitter enables its users to send and read other users' messages, i.e., tweets, each of which is a text-based message up to 140 characters. Users can subscribe to other authors' tweets — this is known as following, and subscribers are known as followers. In this project, the following two Twitter's datasets are used:

- *Twitter's who-follows-whom dataset*, which has the following data format:

USER	FOLLOWER
12	13
12	14
⋮	⋮

Based on the above dataset, we establish a who-follows-whom network. In this network, each node in the graph has links towards its followers, which means its tweets could be received by the followers. In the first phase of the project, we extract a sub-network of the whole Twitter network for our analysis so that the extracted network is within the computation capability of our PCs and still maintain the large-scale nature of the network. The resulting network (referred as the Twitter network) has 59,630 nodes and 1,490,350 edges.

- *Dataset of Twitter posts* (June 2009), which has the following format:

Time of the tweet	2009-06-11 16:59:45
Author	http://twitter.com/ibbored
Tweet content	@amberback #squarespace does? Hot damn. Now I want to win more.
⋮	⋮

This dataset has 18 million tweets. An interesting feature of Twitter is that users use hashtags (#squarespace) to help add tweets to a category (hashtags can occur anywhere in the tweet), and reply to one another using @ sign (such as @amberback). Hence, the propagation of information about a certain topic in Twitter can be monitored by tracking the hash-tags.

2.2 Modularity-based Algorithms to Find Communities

To divide the Twitter network into groups, the network is treated as an undirected network in this project. Three modularity-based algorithms are applied to identify communities in the Twitter network: Clauset-Newman-Moore's algorithm (CNM), Blondel-Guillaume-Lambiotte-Lefebvre's algorithm **(**BGLL), and our modified BGLL algorithm. In this subsection we briefly introduce each of them.

2.2.1 Clauset-Newman-Moore's Algorithm

In 2004, Clauset et al. proposed a hierarchical agglomeration algorithm designed for detecting community structure in very large networks [6]. This algorithm is based on the greedy optimization of the quantity known as modularity. Modularity, denoted by Q, measures whether the division is a good one or not [3]. Consequently, community structures can be found by searching through divisions with high value of Q. The greedy optimization starts with each node being the sole member of a community, repeatedly joins together the two communities whose amalgamation produces the largest increase in Q, and performs the corresponding amalgamation.

2.2.2 BGLL Algorithm

BGLL has linear complexity, which implies it is suitable for large networks. This greedy optimization algorithm is also based on modularity. The algorithm consists of two phases to maximize the modularity of the network. The two phases of the algorithm start with initializing the network with n communities where each node is exactly one community, moving each node to its neighbor's community, finding the community that could make ΔQ maximal, and moving the node into the community found (If no community could make Q increase, then no move is taken). After iterating for all the nodes until the modularity couldn't be increased further by merging, phase I ends and phase II starts. In phase II, the algorithm first creates new network where new nodes are the communities formed in phase I, and the weights between new nodes are the sum of weights between communities formed in phase I (weights inside communities become self-loop). It then returns to phase I, and iterates until Q cannot be increased.

2.2.3 A Modified Version of BGLL Algorithm

We create a modified version of BGLL algorithm, which runs more efficient than the original BGLL. This modified version defines a threshold of the increment of modularity in phase I to avoid inefficient iterations where the gain of the modularity is too small. Once the maximal increment of modularity in one inner loop falls below the threshold, phase I terminates and phase II starts. Although the maximal modularity may not be found at the end of phase I, a sub-optimal modularity in phase II does not affect the final community structures. More importantly, this sub-optimal modularity does save a sizeable computational time, which makes this approach worth exploring. The modified BGLL algorithm is illustrated in Figure 1.

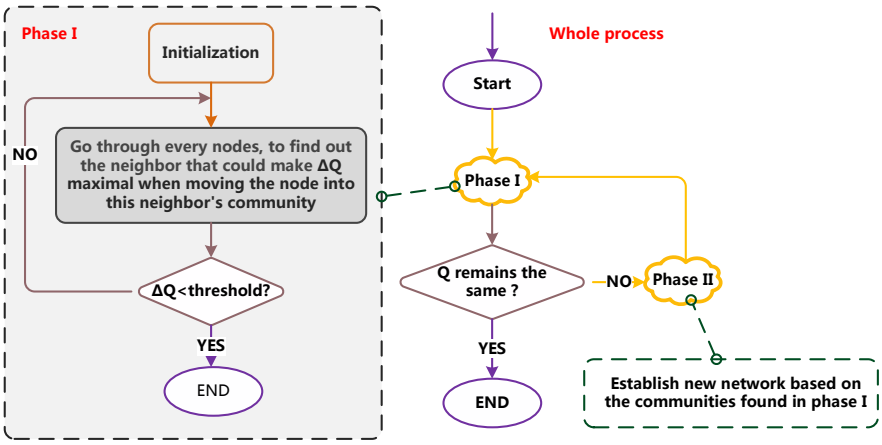


Figure 1. Modified BGLL Algorithm

2.3 Hill Climbing Algorithm to Identify Influential Nodes

We use the hill climbing algorithm to search for influential nodes in the Twitter network. This greedy-based algorithm has an influential function F(S) and starts with an empty set S_0. In each loop, if node v could maximize $F(S_{i-1} \cup \{v\})$, then v will be added into S, until the size of S reaches k, which is our desired number of influencer. In the Twitter's network, every node has an influence set which contains all the nodes it has out-link towards and itself. A function F(S) is defined as the summation of the size of influence set of each node in S.

3. Basic Properties of the Twitter Network

In this section, we describe the basic properties of Twitter's who-follows-whom network by providing the degree distribution, K-core decomposition, and other statistics.

3.1 Degree Distribution

Figure 2 and Figure 3 shows the in/out degree distribution and corresponding complementary cumulative distribution function (CCDF) of the network. Evidently, the degree distributions follow power law as other real world networks do. Using maximum likelihood estimate (MLE), we obtain the parameter of the power law, $\alpha = 1.5$, approximately.

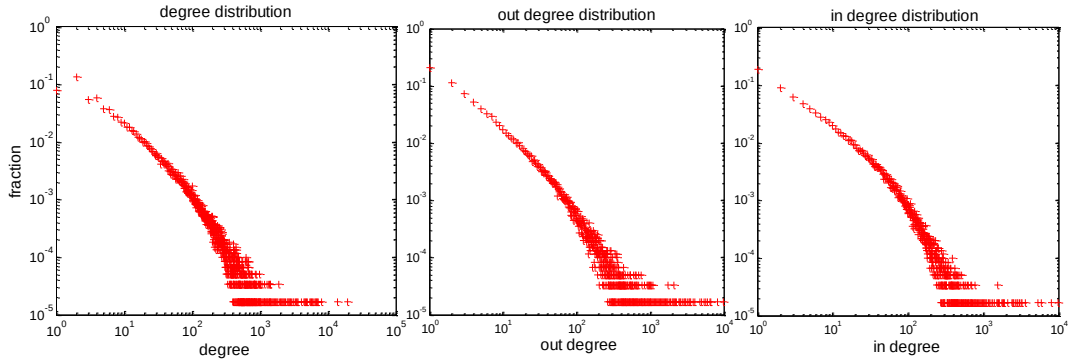


Figure 2. Degree distribution of Twitter's who-follows-whom network

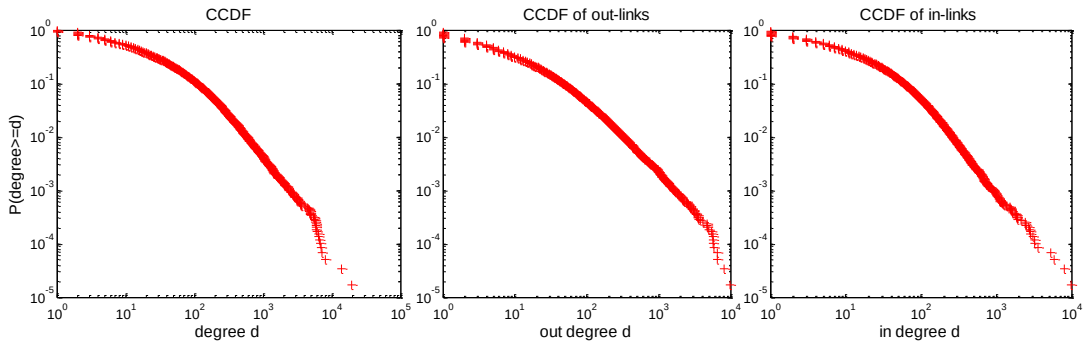


Figure 3. Complementary cumulative distribution function

3.2 K-Core Decomposition

Figure 4 shows the K-core decomposition of Twitter's who-follows-whom network. We observe that as k increases, the number of nodes remaining in k-cores decreases exponentially, i.e. this process also follows a power law. When $k \geq 180$, no k-core remains in the network. This indicates

that the network has tightly connected components that could be used for large-scale information propagation.

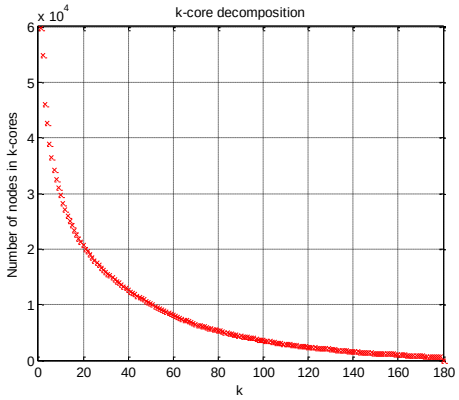


Figure 4. K-Core Decomposition

3.3 Other Statistics

Other statistics of the network are listed in Table 1.

Nodes	Edges	Average clustering coefficient	Average short path	Size of largest connected component
59,630	1,490,350	0.294	3.2919	47,951

Table 1. Statistics of Twitter's who-follows-whom network

4. Community Detection in the Twitter Network

4.1 Comparison of Three Modularity-based Algorithms

In this section, the performances of the three algorithms (CNM, original BGLL, and modified version of BGLL) in detecting communities in different sizes of Twitter networks are compared based on the final modularity and computing time. Results are shown in Table 2 and Table 3.

Network Size	2.9K nodes 15.6K edges	10K nodes 100K edges	12K nodes 170K edges	60K nodes 1.5M edges
CNM	0.329742	0.330634	0.339299	0.341022
BGLL	0.375919	0.377361	0.384498	0.382019
Modified BGLL (threshold h=0.0001)	0.375919	0.377335	0.384327	0.381670

Table 2. Final modularity found by the three algorithms

Network Size	2.9K nodes 15.6K edges	10K nodes 100K edges	12K nodes 170K edges	60K nodes 1.5M edges
CNM	10s	179s	242s	1769s
BGLL	5s	32s	49s	347s
Modified BGLL (threshold h=0.0001)	5s	25s	36s	275s

Table 3. Running time of the three algorithms

From the above results, we can see that compared to CNM, BGLL achieves a higher modularity using less time. Hence BGLL performs better on large networks. On the other hand, our modified BGLL reaches approximately the same modularity using less time than BGLL. This result becomes more remarkable as the size of the network increases as Table 3 indicates. Therefore, the modified version of BGLL does improve the computing speed compared to BGLL without the risk of reducing the modularity.

4.2 Reduction in Modularity Using Modified BGLL with Different Thresholds

In this subsection, we investigate the reduction in modularity (ΔQ) and saving of computing time by running modified BGLL with different thresholds (h) in different networks. As can be seen from the left panel of Figure 5, when h increases, the reduction in modularity by modified BGLL grows linearly. The right panel of Figure 5 shows that when h increases the saving of computing time (%) also increases. As a matter of fact, a large ΔQ is not desired since it represents the fact that the modularity obtained by modified BGLL is considerably smaller than the modularity obtained by BGLL. On the other hand, a large saving of computing time is desired. Consequently, the choice of h is a tradeoff between the risk of reducing modularity too much and the risk of increasing computing time. Given the three networks considered in this section, it is found that saving of time increases fastest when h is in the interval (0, 0.0005), so 0.0005 is a reasonable choice because modularity only decreases by a small amount (less than 0.0005) at that point while saving of time is remarkable (30%~40%).

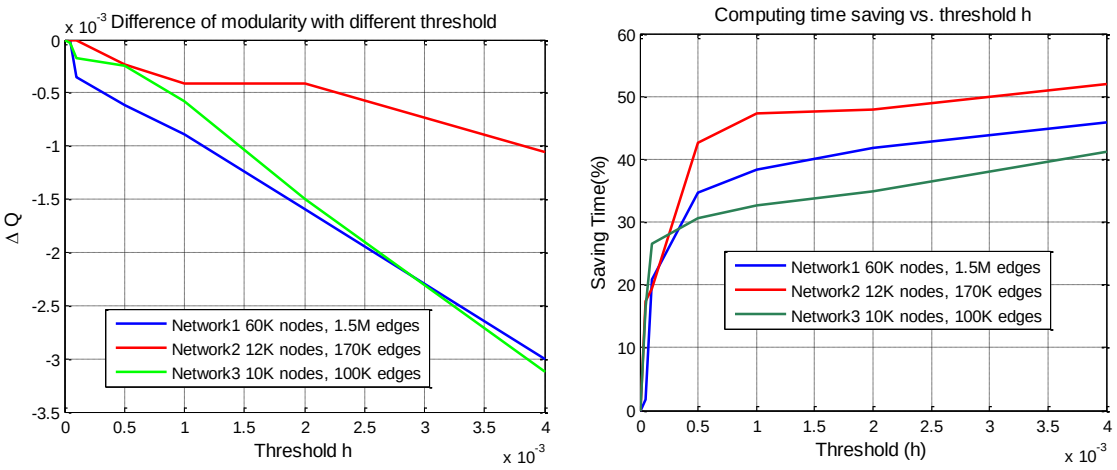


Figure 5. Performance of modified BGLL with different threshold h

4.3 Community Detected by Modified BGLL

Applying the modified BGLL (h=0.0005) to the Twitter network, we obtain 930 communities. The connected communities are illustrated in the left panel of Figure 6, where each node is a community. The right panel of Figure 6 shows the network of a single community. We observe that every single community has dense inner-links and sparse out-links, implying that information can quickly spread within a community but might be hard to spread outside the community.

4.4 Verifying Detected Community by Examining Network Conductance

After the communities are obtained, it is necessary to examine the conductance of the network to validate the communities we found. The conductance is defined by:

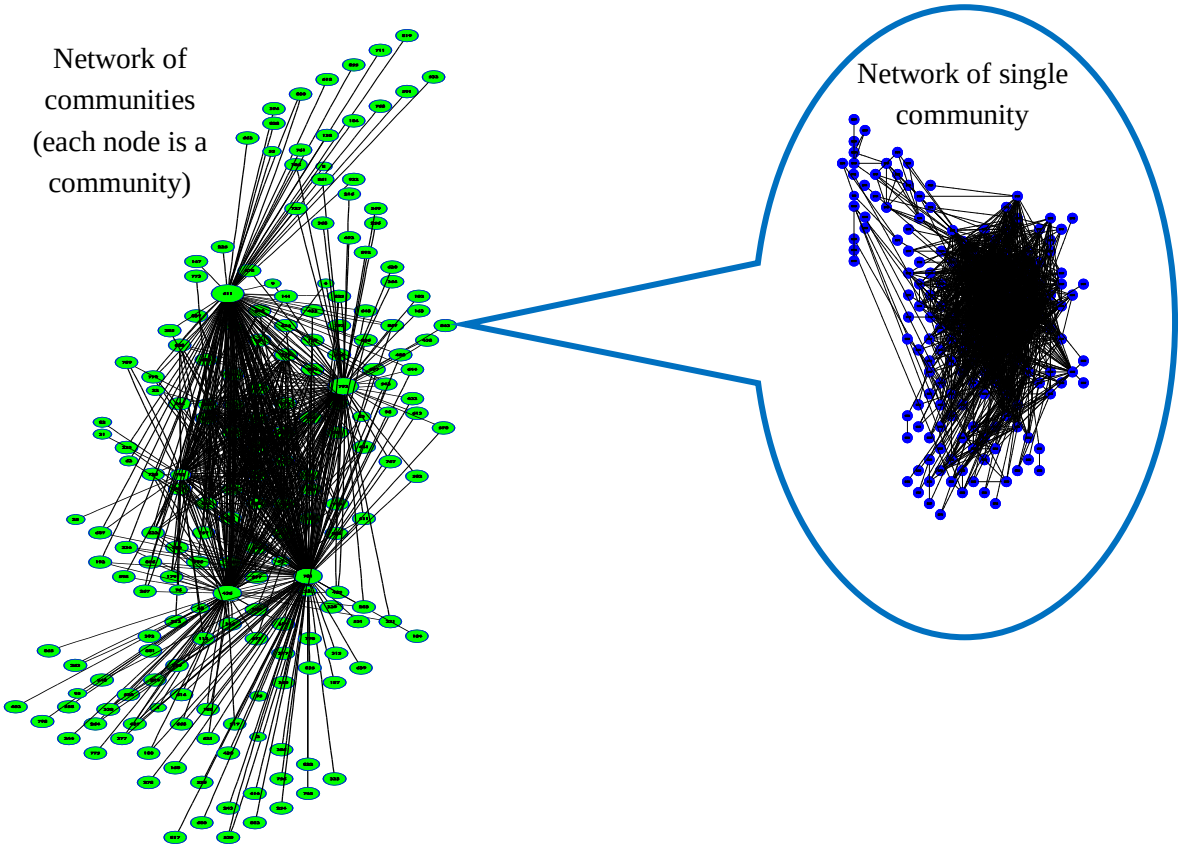


Figure 6. Connected part of the community in the Twitter network

$$\phi(C) = \frac{\sum_{i \in C, j \notin C} A_{ij}}{\min\{A(C), A(\bar{C})\}} \quad \text{, where} \quad A(C) = \sum_{i \in C} \sum_{j \in V} A_{ij}$$

Figure 7 plots the network conductance against the size of a community. It can be seen that as the size of a community increases, the conductance tends to increase at first and stabilize after the community becomes sufficiently large. This result demonstrates that the communities we found have bounded conductance and hence our modified BGLL is effective in founding the communities in the Twitter network.

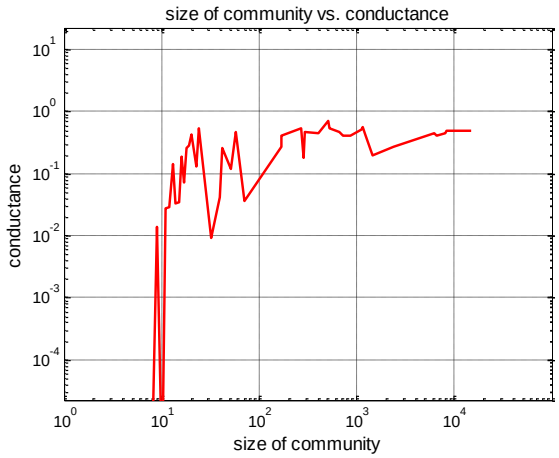


Figure 7. Conductance of communities in Twitter network

5. Information Propagation in the Twitter Network

In this section, we explore the pattern of information propagation in the Twitter network using the dataset of Twitter posts, identify the most influential nodes in the network, and investigate how different starting nodes affect the propagation process in a community as well as in the entire network.

5.1 Types of Topic in the Twitter Network

As Twitter's users use hashtags to help adding tweets to a category, patterns of propagations of information can be observed by tracking certain hash-tags. A hashtag-mention distribution of this network is shown in Figure 8. It can be seen that the hashtag-mention distribution follows power law with parameter α=2.07. We observe that fewer topics have large number of mentions, indicating there exist different patterns of propagation of different topics.

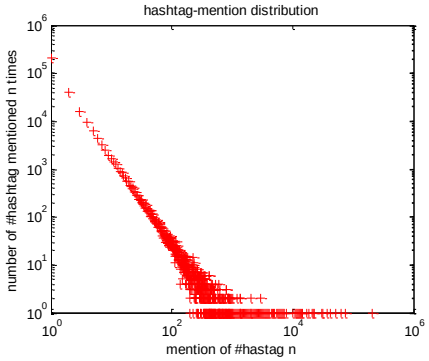
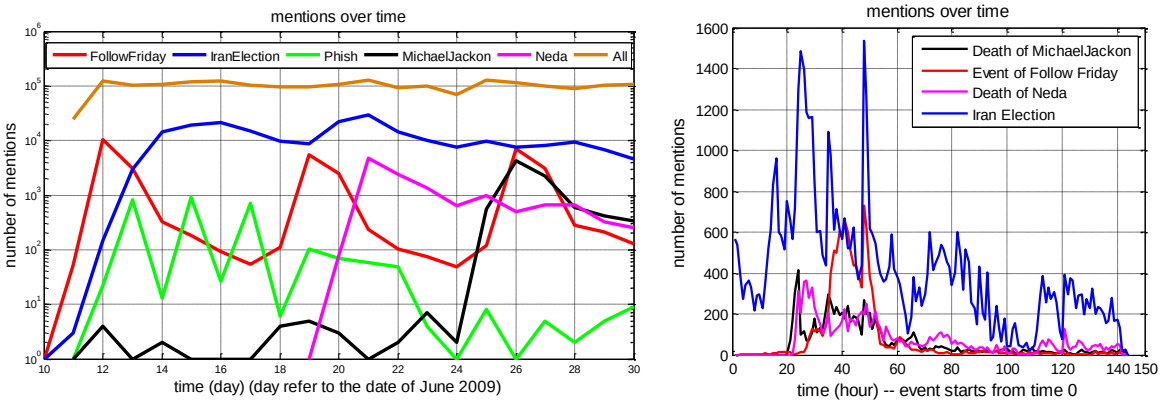


Figure 8. Conductance of communities in Twitter network

Figure 9(a) illustrates different propagation patterns of five different topics. We observe that the topics can be categorized in three types: long-term topic, periodic topic, and sudden topic. As shown in Figure 9(a), long-term topics, such as #IranElection, always have high mentions; periodic topics, such as #FollowFriday, have high mentions recursively at some particular time in a period (in this example, every Friday); sudden topics, such as the death of Michael Jackson or death of Neda, obtain a significant number of mentions in a short time and then disappear rapidly, as shown in Figure 9(b).



(a) (b)

Figure 9. (a) mentions over time (day) of different types of topics
(b) mentions over time (hour) of some sudden topics

In particular, a long-term topic can be viewed as continuous recurrence of sudden topics, while a periodic topic can be viewed as periodic recurrence of sudden topics, as shown in Figure 9(b). Therefore, since any type of topic can be seen as a combination of sudden topics, studying the propagation properties of a sudden topic allows us to understand the propagation properties of other types of topics as well.

5.2 Identify Most Influential Nodes

In the propagation process of a sudden topic, there exists an initial propagation set of nodes, called starting nodes that spread the information in the first step. Ideally, there exists a certain set of nodes (influential nodes) which can spread the information farthest. To investigate how these influential nodes affect the propagation of different types of topics in the Twitter network, we need to identify these starting nodes in the first place.

To do so, the hill climbing algorithm (Section 2.3) is used to find the most influential nodes in a Twitter network (2.9K nodes, 15.6K edges). To validate the influential nodes found by this method, a following simulation is performed: assume 100 influential nodes start a propagation process, and assume every node receiving the information spreads the news to each node in its influence set. Then how far the news is spread can be evaluated by the number of nodes that are reached eventually. By comparing this case to the cases where 100 random nodes or 100 high-outdegree nodes (nodes picked by the order of outdegree) are selected as starting nodes, the influential nodes found by the hill climbing algorithm can be verified. The results are shown in Table 4. Evidently, the influential nodes found by hill climbing method spread information most widely. Notice that this is an ideal case since we assume each node deterministically spreads the topic to nodes in its influence set. A more realistic case is considered in Section 5.3.

	Influential nodes	High-outdegree nodes	Random nodes
Nodes influenced (%)	0.841537	0.784221	0.815105

Table 4. Comparison of nodes influenced by three kinds of starting nodes (k=100)

5.3 Information Propagation in a Community vs. in an Entire Twitter Network

In this section, we explore how to choose starting nodes that can spread a sudden topic fastest and most widely in a community as well as in an entire Twitter network.

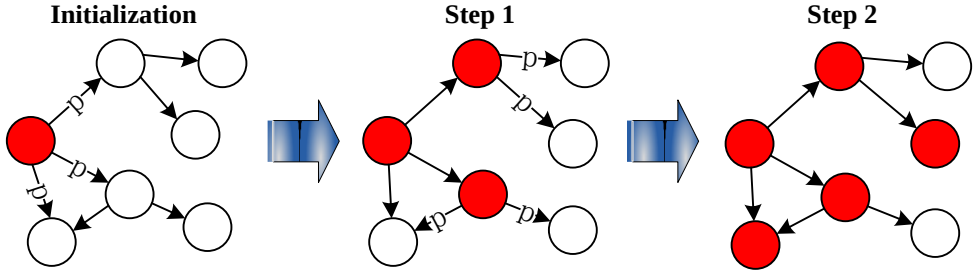


Figure 10. Simulation of the information propagation in the Twitter network

To compare the performance of different starting nodes, another simulation is conducted. As shown in Figure 10, initially several nodes are set as starting nodes. At each step, these

influenced nodes spread information to their out-neighbors. Each node receiving the news has probability p to be influenced and spreads to its out-neighbors, where p depends on different types of topic. For example, p is large when the topic is popular, whereas p is small when the topic is less popular.

5.3.1 Propagation in a Community

We run the simulation in a community detected in Section 4.3 (168 nodes, 2879 edges) with 10 starting nodes (k=10). We compare the performances of three different types of starting nodes: influential nodes found by hill climbing method, high-outdegree nodes selected in the order of outdegree, and randomly selected nodes. The number of people affected (%) in a community against the step of propagation is shown in Figure 11, and the number of people affected (%) in a community against the number of starting nodes is shown in Figure 12. We observe that if the topic is popular (p=0.2), the information can always spread widely regardless of the type of the starting nodes, as Figure 12 shows. However, the influential nodes found by hill climbing method can spread the topic faster as it takes fewer steps to propagate, as shown in Figure 11. On the other hand, if the topic is not popular (p=0.01), the high-outdegree nodes always propagate the information fastest and most widely.

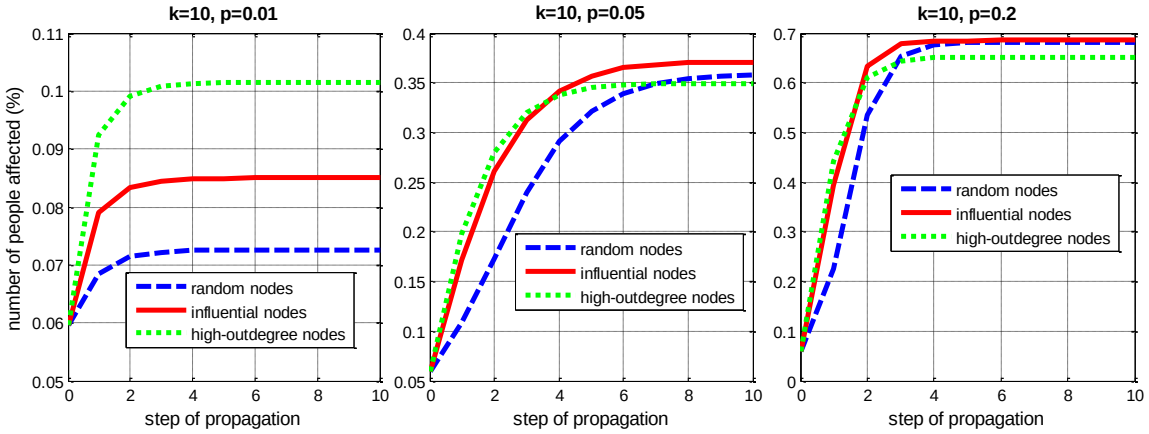


Figure 11. Number of people affected (%) in a community against the step of propagation

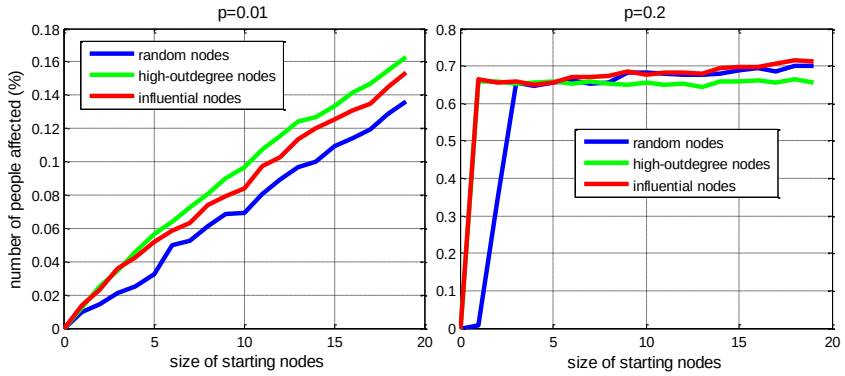


Figure 12. Number of people affected (%) in a community against the number of starting nodes

5.3.2 Propagation in an Entire Twitter Network

Next we run the simulation in an entire Twitter network (2.9K nodes, 15.6K edges), and the

number of starting nodes is set to be 100 (k=100). We also compare the performances of three different types of starting nodes mentioned above. The results are shown in Figure 13 and Figure 14. We observe that if the topic is popular (p=0.2), the influential nodes can spread the topic farthest and fastest; if the news is not popular (p=0.01), the high-outdegree nodes have the best performance in propagating the topic widely and quickly.

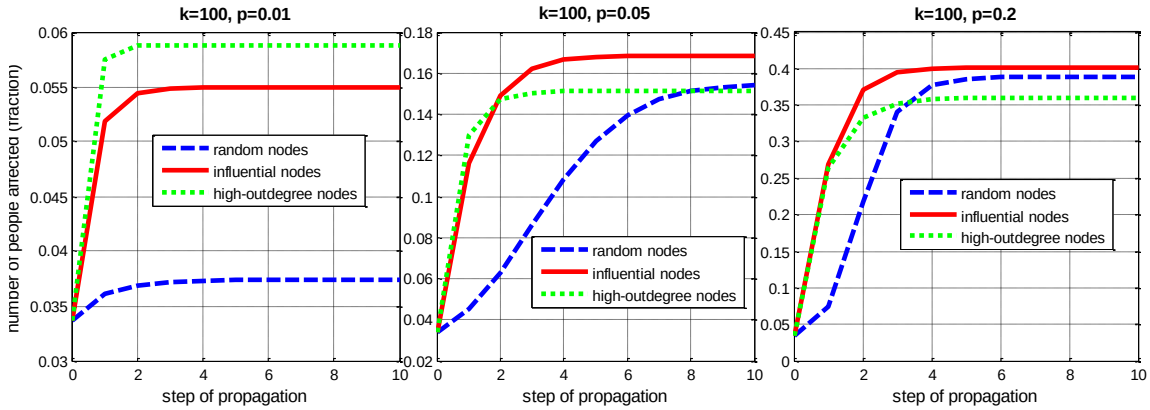


Figure 13. Number of people affected (%) against the step of propagation in an entire network

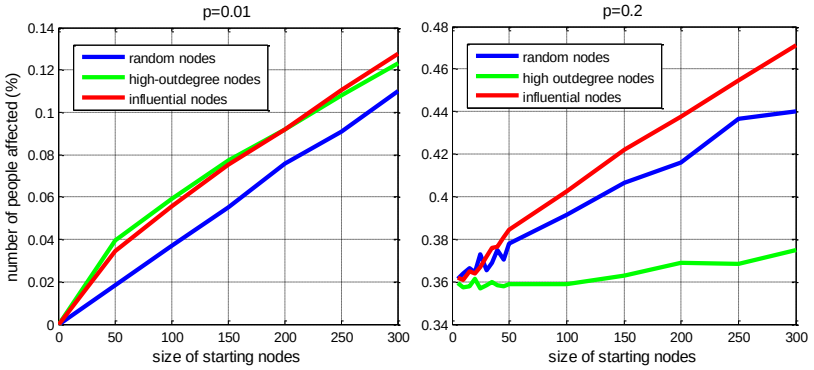


Figure 14. Number of people affected (%) against the number of starting nodes in an entire network

6. Conclusion and Future Work

The first part of this project focuses on community detection in the Twitter network. Specifically, we modify the BGLL algorithm by setting a threshold of the gain of modularity in phase I of the algorithms to make it work more efficiently while maintaining the resulting communities with high modularity. After the tradeoff analysis between decreasing computing time and maximizing modularity, we choose a moderate value of threshold h=0.0005. Conductances of the resulting communities found by the modified BGLL converge and are upper-bounded, which implies this algorithm works well on the Twitter network.

The second part of this project emphasizes on information propagation in the Twitter network. In particular, we investigate the propagation patterns of different types of topic spread in the Twitter network and conclude that any topic can be viewed as a combination of sudden topics. Consequently, we focus on the spread of sudden topics in a community and in an entire network by running simulations with three types of starting nodes (influential nodes found by hill climbing, high-outdegree nodes and random nodes). The results tell us that not only the different

types of starting nodes could affect the propagation progress; the different types of information also could influence the pattern of propagation.

Our future research revenue includes (1) investigating propagation patterns in a real Twitter network to verify the observations obtained from our simulations; and (2) exploring how an information cascade in the Twitter network behaves and how to predict the pattern of cascades given a starting set of nodes.

Acknowledgment

We thank Prof. Leskovec, Sonali, and Sadine for helpful discussions and suggestions to this project as well as providing us the Twitter network dataset. We would also like to thank Prof. Leskovec and all the course assistants for bringing us such an amazing learning journey in this quarter.

References

1. Blondel, V.; Guillaume, J.; Lambiotte, R; Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, IOP Publishing*, 2008.
2. Fortunato, S. Community detection in graphs. *Physics Reports, Elsevier,* 2010, (486):75-174
3. Newman, M. Analysis of weighted networks. *Physical Review E, APS*, 2004, 70.
4. Leskovec, J.;McGlohon, M.; Faloutsos, C.; Glance, N.; Hurst, M. Information Propagation and Network Evolution on the Web. *DA Project, Machine Learning Dept. Carnegie Mellon University*.
5. Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; Glance, N. Cost-effective Outbreak Detection in Networks. *KDD'07, August 12–15, 2007, San Jose, California, USA.*
6. Clauset, A.; Newman, M.; Moore, C. Finding community structure in very large networks. *Physical Review E, APS,* 2004, *70*, 66111