Detection of Head Node Using Graph Mining

Mansi Patel
University at Albany, SUNY
1400 Washington Avenue,
Albany, New York 12222
mnpatel@albany.edu

Dishaben Patel
University at Albany, SUNY
1400 Washington Avenue,
Albany, New York 12222
dpatel3@albany.edu

Bhavik Patel
University at Albany, SUNY
1400 Washington Avenue,
Albany, New York 12222
bpatel2@albany.edu

ABSTRACT

In today's world, information via social media circulates rapidly. But no one can easily assure whether the information is substantiated or it is a hoax. So, the detection of such fake information in social networks is beneficial to society. Moreover, it is equally important to bring the epicenter of this fake information chain in the network. Based on community detection, a fast and efficient method for detecting the source of such fake information thread is proposed. Our approach is, from all the individual communities, find the source nodes of each individual communities and conclude with one source node that us resulted in highest number of times to be source of communities as a susceptible one. Thus by our method we could find one particular head node which is likely to be source or head in highest number of communities. And from this we could find the network structure which seems common in past. Therefore, we could suspect that particular head to be illicit to other activities, if any one of its diffused community is felonious.

KEYWORDS

Community Detection

Head Node/ Source Node

Label

Social networks

1 INTRODUCTION

One of the most important tasks when studying information from social networks is that of identifying network communities and source of the information. Fundamentally, communities allow us to reach the source of the information. For an example, the news of having a fire in Seattle, is currently widely concerned with the media. But is this concern worth? Is this news real? Or is this fake. If it is fake, then who has tried to spread this? Who wanted to prank it? So, all the answers could be solved when we will dig out the community related to this news. And from this community, we will dig out the generator of this news. If this generator is same for many other doubtful communities, than this helps the social sites to take action against the news and the very same source behind the news. So, our approach finds the source which is head node of many communities. And by this find the back bone structure formed in the past.

Identifying network communities can be viewed as a problem of clustering a set of nodes into communities linked via topic in trend as edges. That is, nodes can be individual users, which are connected by topic discussed by them on a social platform. Performing our proposed community detection algorithm, it will concise our search for source node. Also, the main thing to point out is our community detection algorithms are aiming to find communities based on the network structure, e.g., to find groups of nodes that are connected densely based on some topic or information. Also, studying these communities will give us information to overlapping communities. Overlapping communities are formed when one or more user of one community is linked with other community. Thus, at the present situation, it ignores node attributes such as user information and so on. Finally, with that limited network, we can easily present out the source node.

2 DATASET

Data description

Twitter is a treasure trove of data and there are plenty of interesting things and there are plenty of things one can discover. One can pull various data structures from twitter: tweets, user profiles, user friends and followers, what's trending etc.

We obtained Training dataset from Twitter using the hyperlink: http://help.sentiment140.com; created by Alec Go, Richa Bhayani, and Lei Huang, from Stanford University. Dataset (in csv format) contains tweets with the following information:

- Polarity of the tweet(0= negative, 2= neutral, 4= positive)
- Id of the tweet
- Data of the tweet
- User Name
- Text of the tweet.

The tweets in the data set are from the time period between: April 6, 2009 to June 25, 2009.
Further, as the sub part of the project is based on detecting the head nodes of each communities, after analyzing the dataset, we have removed the unnecessary data from the dataset, over here we have obliterated the field: polarity of the tweet. After processing, now dataset attributes appears as shown in below table. **Error! Reference source not found.**.

No.	Column Name	Format Example	Description
1.	Tweet_id	32327382	Id of tweet
2.	Tweet_time	Sat May 16 23:58:44 UTC 2009	timestamp of tweet
3.	User_name	rocketscience575	Username who tweeted
4.	Tweet	Obama is visiting istanbul today, therefore all main roads have been closed cause and effect!!!	Tweet text

During the project proposal, we stated about using the dataset from Standford Snap. But now, dataset from this link is no longer available, so we switched to another dataset (Alec Go).

3 EXPERIMENTAL AND COMPUTATIONAL DETAILS

3.1 Data Preprocessing

The dataset was to be processed before experimenting. As mentioned in above section, it had large tweets. We had to apply data mining algorithm such that an important word as label is selected from the tweeted text.

Example 1: User A had tweeted: "@me_to_you I think I like him by now, I need his hug ;) ". This tweet contains important label as "hug", where other words are under stop words in some or the other way.

User B had tweeted: "need a hug, no matter from whom…sob sob☹", where hug is main label ignoring the stop words.

User C had tweeted: "@kapil_sssharma wonderful performance…☺ ☺"

This was achieved by performing that data mining algorithm which takes the normal tweet text data and returns most frequently used words like "hug" label is used twice in above examples. Moreover, those are later on to be truncated to hundreds of tweet labels; considered as a test dataset.

Error! Reference source not found.

Label	User	Tweet
San Francisco	schuyler	just landed at San Francisco
jquery	dcostalis	jquery is my new best friend.
exam	jvici0us	History exam studying ugh.

Thus, in data mining portion, actual logic behind the dataset is to find a label word from the tweet as shown in **Error! Reference source not found.**.

So, by using the data of twitter users and its label, along with its respective timestamps, we have generated a network with help of Gephi tool. This was achieved by running a python script which generates csv files of nodes and edges from the test dataset. We had chosen an approach: Consider users as nodes and interlinks between them as edges to user nodes via common topic tweeted by those users. The algorithm use to do so, is shown in

Figure 1 Data Preprocessing Algorithm.

Example 2: User A tweeted on topic labelled as: "US Election" and User B had also twitted on topic labelled as: "US Election".

Now, we would be having an edge between User A and User B linked by "US Election" as shown in **Figure 2 Network for Example** .

```
Graph G
 for all i in data.users:
        G.add_node(i)
        Unique_topic_tweet.append(data_topic)
for all i in G.nodes:
        for all j in G.nodes:
                if( i.notequales(j) and data.tweet(i).equals(data.tweet(j)))
                        G.add_edge(G.node[i], G.node[j])
```

Figure 1 Data Preprocessing Algorithm

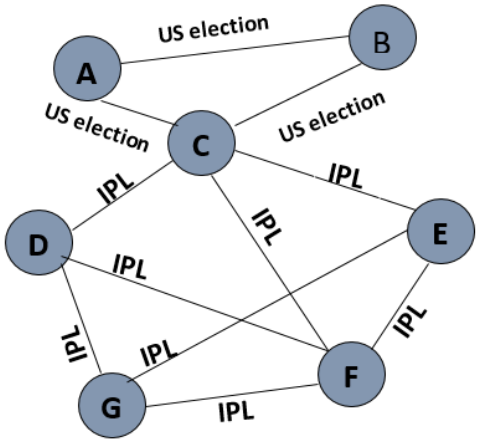


Figure 2 Network for Example 2

3.2 Community Detection

Community detection is the initial approach for our main motive to get the source node. Community detection is performed to confine our search of source node. Moreover, we could analyze common overlapping communities. Overlapping communities would help us to get information of nodes for particular case where those nodes are head nodes of multiple communities.

Therefore, we need to find community and its respective source nodes. Once, we get the target community, our search would have a confined precise boundary to work upon. Community detection is mainly based on finding a group of users which shares same topic (label in our case) among their tweets.

Let us get back to **Figure 2 Network for Example** , where we have users C, D, E, F and G. In the past, they had tweeted on the same label IPL (Indian Premier League), professional Twenty20 cricket league in India contested during April and Ma`5r4y of every year by franchise teams representing Indian cities. So, we will form a community for this IPL group between these users. All users are treated as nodes, who had tweeted for this label: "IPL". They will be connected by an edge with all the other users who had tweeted with this very same label.

Also, from the same **Figure 2 Network for Example** , we have User C, as a common user such that, he has tweeted for both the labels: "IPL" and "US election". Thus, it is a common node for both network communities like in **Figure 3 Community formed for figure 2**. Moreover, in case of any chance, if the same User C has tweeted in the earliest of time compare to other users for both labels, then according to our approach we got it as our source node for both labels. And if by any chance it had hoaxed regarding "US election" for being its source node, then it is likely to suspect him for other all tweets where he was source. So, here is the point where our approach comes into the picture and plays a major role.

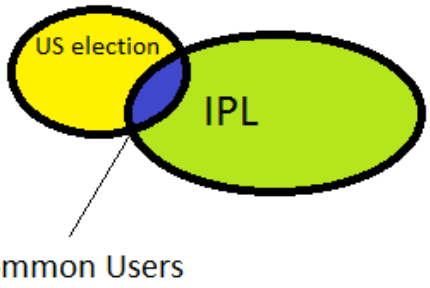


Figure 4 Community formed for figure 2

4 RESULTS AND DISCUSSION

We gathered some results after performing some experiments to find head nodes from the communities. Moreover, after applying community detection to test dataset, we could perform certain network analysis. The main thing to note down is that all the data set used in this report is totally upon the test dataset (part of actual dataset, which was resulted after data preprocessing).

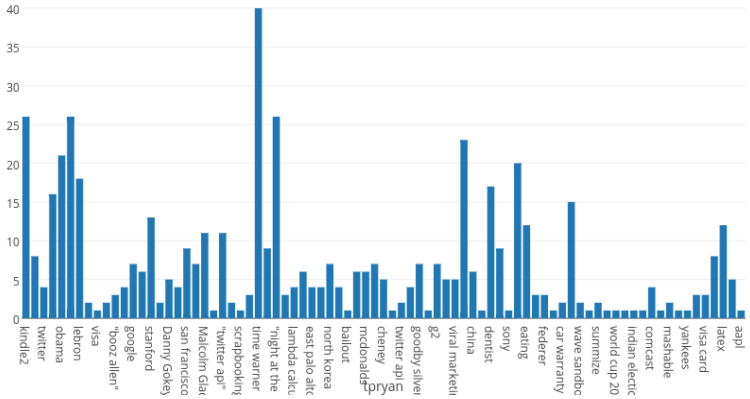


Figure 5 Histogram plotting number of times the particular label, tweeted by users

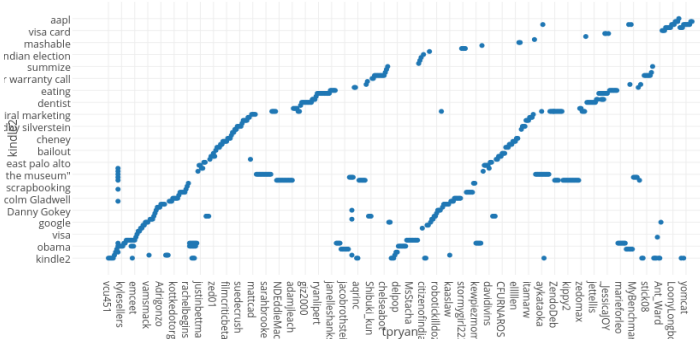


Figure 6 Scatter plot showing User to Tweet Label Relationship

Figure 7 Word Cloud for Tweet Labels

In **Error! Reference source not found.**, we can see how many times the particular label has been tweeted by all users. Similarly, **Error! Reference source not found.**, and it has detail information that which user has tweeted on which label.

Now coming back to our community networks, **Figure 8** shows communities of those tweet labels such that users are treated as nodes and edge label is the tweet label. More the denser community in center, more number of users are related to that community. And the peripheral area of figure shows the less tweeted topic by users.

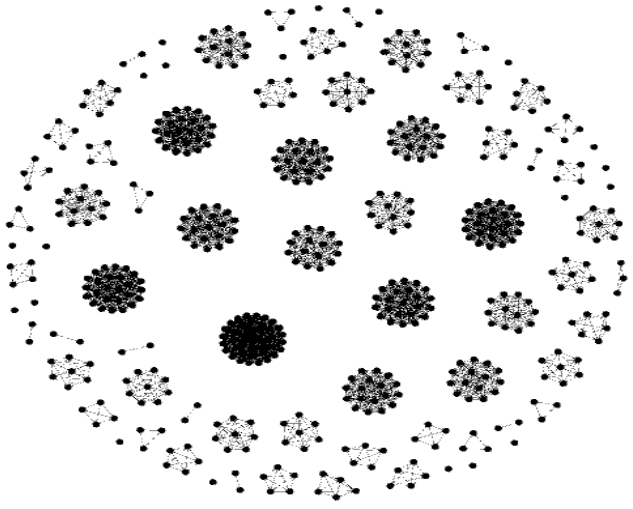


Figure 8 Community Network: force Atlas layout in Gephi Tool.

Further, we also came across overlapping communities where user has tweeted for two different topics and therefore he is node for two different community network. This user helps in finding the common structure followed by his friends or his follower friends. And by this we could trace that sequence of users to suspect if any one of them was under fraud tweet scandal. This concept for overlapping communities is clearly seen in **Figure 8(a) and 8(b).**

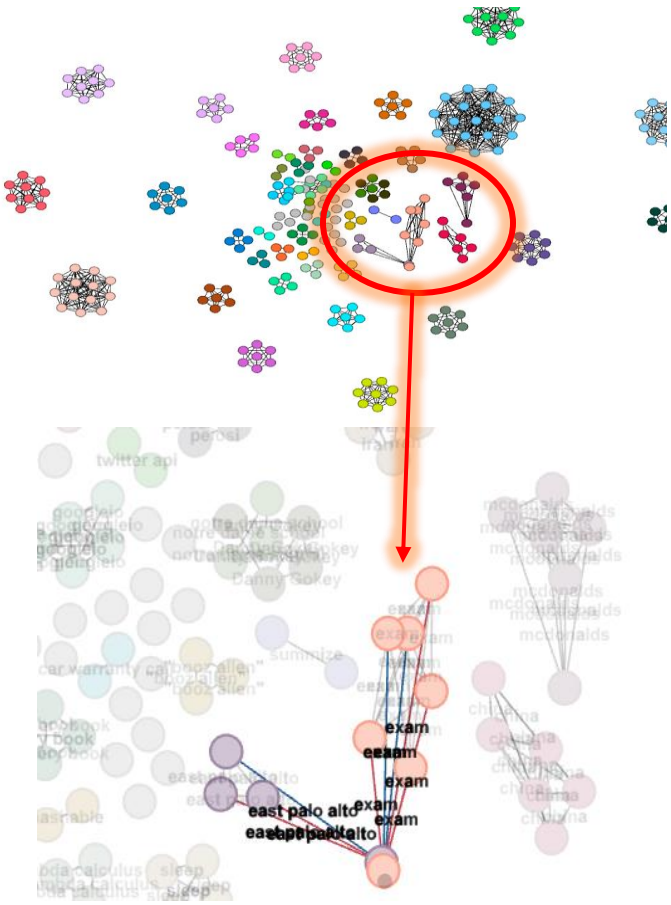


Figure 8(a) Overlapping Communities
Figure 8(b) Enlarged Overlapping Communities along with its specific Tweet Label

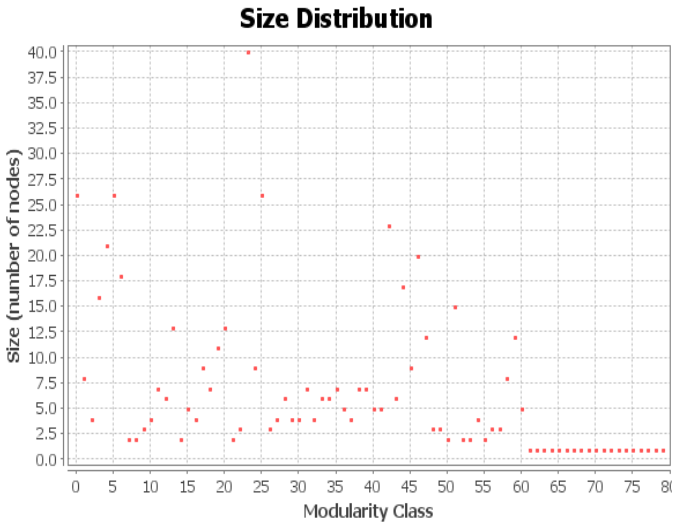
In the very same above **Figure 8(a) and 8(b)**, we came up with this network results by using Gephi Tool and its Force Atlas Layout. From this figure, with edge Label (Tweet label) as *"east palo alto"* and *"exam"* from the same user of our test dataset.

Error! Reference source not found.

Randomize	On
Use edge Weights	On
Resolution	1.0

Error! Reference source not found.

Modularity	0.917
Modularity with resolution	0.917
Number of Communities	80



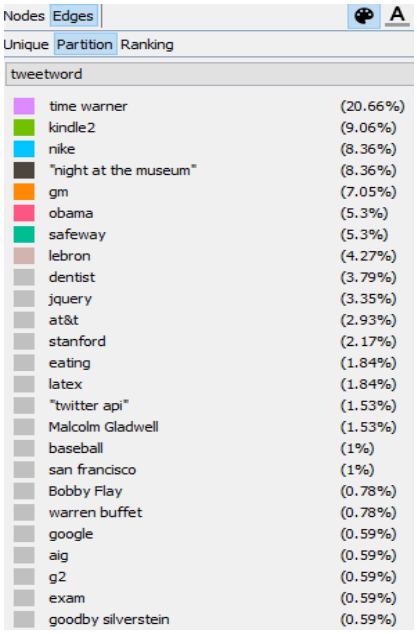


Figure 9(a) Size Distribution of Modularity Report
Figure 9(b) Edges with respective tweet labels percent

So, we could find all communities count and the modularity of network as quite nice count by **Table 3 and 4** and **Figure 9(a) Size Distribution of Modularity Report** is the result of modularity report of our network form. Also, we experimented out the nodes which are head nodes from all communities. We also found a common sequential structure of nodes from all those communities successfully.

4 Conclusion and Future Work

In a nutshell, Community detection is the base for finding the head node of different communities among the entire dataset. So, first of all we detected communities among the users after finding out respective labels of tweets from the dataset. Then using Gephi tool we analyzed the networks created from these communities. Later on we found nodes which roles as to be source node in many of those communities. Also, we could find a sequence of nodes which seems to be consistent in all communities. All this tasks are done by python scripts.

The research work presented in this paper has many applications. Like technology is evolving, social media is too evolving at a great speed and so necessary to find fraud news and its source. Also, in cancer detection or terrorist networks it can be used.

Moreover, this paper can be carried further in future by trying different dataset, or by using same dataset in higher configuration server.

5 Work Distribution

Mansi Patel: Implementation of Project (All Coding), Documentation of Reports, Presentation slides, Network Analysis

Disha Patel: Dataset Analysis, Literature Survey, Documentation

Bhavik Patel: Proposed Approach, Literature Survey, Presentation slides, Network Analysis

6 References

[1] Word Cloud: http://tagcrowd.com/
[2] Yang, Jaewon, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes." Data Mining (ICDM), 2013 IEEE 13th international conference on. IEEE, 2013.
[3] Pan, Y., Li, D. H., Liu, J. G., & Liang, J. Z. (2010). Detecting community structure in complex networks via node similarity. Physica A: Statistical Mechanics and its Applications, 389(14), 2849-2857.
[4] Gephi Tool: https://gephi.org/
[5] To plot results: https://plot.ly/
[6] Fortunato, Santo. "Community detection in graphs." Physics reports 486.3 (2010): 75-174.
[7] Word Cloud generator: http://tagcrowd.com/
[8] Twitter Dataset (2009 to 2014): http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip