Manipulation Detection in Online Discussion Forums
Final Report

Himanshi Sinha (12202) ME/CSA Lakshmi Bansal (13014) ME/CSA
Poonam Kashyap (12409) ME/CSA Sweta Sharma (12392) ME/CSA

April 16, 2016

1 Introduction

We tackle the problem of strategic manipulation by individuals participating in an online discussion forum. In these discussion forums, there are two special features, which highlights the genuineness of a post submitted by an individual, namely upvote and follow.These are also the features through which the forum incentivizes its users, as more upvotes and followers make the user an expert in that system. In such systems, a group of users can form a coalition and follow and upvote the contents posted by the members of that group, thus increasing the expert rank of the group members. Such manipulation can lead to the formation of fake experts in such discussion forums thus decreasing its quality. In order to tackle the above problem, we use social networks inducted on the piazza discussion forum, and propose modularity as the measure of manipulation detection in order to handle such anomalies.

2 Problem Definition

We have reduced the problem of manipulation detection to the problem of community detection in social networks because the "who follows whom" and "who upvotes whom" network can be seen as the social connections between the members of the discussion forum. This in turn, implies that we have induced a social network and we can use community detection algorithms on it.
Social Networks. A *social network* with n individuals and m social ties can be denoted as $G(V, E)$, where V is the set of nodes, $|V| = n$, and E is the set of undirected relationships, $E \subseteq V$xV , $|E| = m$. A social network is also referred to herein as a graph.
Communities. Non-overlapping communities are not confined to a graph partition, and clusters which incompletely cover the graph are usually more desirable. Here we define the communities as a list of non-empty node subsets: Coms = $\{V_1^{'}, ..., V_{cn}^{'}\}$, where $\bigcup_{i=1}^{cn} V_i^{'} \subseteq V$, and cn is the total number of communities. Please note *Coms* should try to satisfy $V_i^{'} \cap V_j^{'} = \phi$. A community is also referred to as a cluster or a part.
PROBLEM DEFINITION 1. Generally, given a network $G(V, E)$, the community detection problem aims at finding the optimal community assignment $R(Coms, Outs)$ from G, such that (1)$Coms \cap Outs = \phi$ and (2)$Coms \cup Outs = V$. Herein the optimal assignment refers to closely connected groups of nodes (Coms) and a moderate number of disparate outliers (Outs).

This problem definition is working for undirected networks and the induced networks we have are directed. So we, convert these social networks to the undirected one by following techniques:
A+A':
If given graph is A then we use A+A' as the undirected graph which means we are considering all directed edges as undirected.
AA'+A'A(triadic closure):

It can be thought of as if two people in a social network have a friend in common, then there is an increased likelihood that they will become friends themselves at some point in the future. Triadic closure is intuitively very natural, and essentially everyone can find examples from their own experience. One reason why B and C are more likely to become friends, when they have a common friend A, is simply based on the opportunity for B and C to meet: if A spends time with both B and C, then there is an increased chance that they will end up knowing each other and potentially becoming friends.

These two methods have no notion of similarity in them. They are complementary to each other. First one can be thought of working on given network and the second one can be thought of as the network which may evolve over time

3 Challenges

- Choosing the appropriate measure for community detection.
- Choosing the most effective algorithm for community detection.
- Differentiate between genuine and manipulative coalitions.
- To penalize the members in such a way so as to discourage them to not form such coalitions.

4 Measures

Modularity: Fraction of the edges that fall within the given groups minus the expected such fraction, if edges were distributed at random.

Clustering Coefficient: Measure of the degree to which nodes in a graph tends to cluster together.It shows how close the neighbours of a vertex are, to being a clique.High CC means that the connection inside the communities is dense.

$$CC = \frac{1}{cn}\sum_{i=1}^{cn}\left(\frac{1}{|C_i|}\sum_{v\in C_i}\frac{2|\{e_{ts}:v_t,v_s\in N(v)\cap C_i,e_{ts}\in E\}|}{d(v)(d(v)-1)}\right)$$

5 Maximal K-mutual Friends(MKMF)[2]

Maximal k-Mutual-Friends (M-KMF) [2] algorithm incrementally filters out the connections by the number of mutual friends between nodes to let the communities spontaneously emerge. It is a *Direct-Partitioning* method. This algorithim has very high CC [3]

Definitions:
Mutual-Friend: If a social actor 'a' is a common friend of two other social actors 'u' and 'v' then 'a' is a mutual friend of 'u' and 'v'.
K-mutual friend: saying 'u' and 'v' have k-mutual friends means there are other 'k' number of social actors who are mutual friends of 'u' and 'v'.
Maximal k-mutual-friend Subgraph: A maximal k-mutual-friend subgraph is a k-mutual-friend subgraph that is not a proper subgraph of any other k-mutual-friend subgraph.

Given a social graph G and the parameter k, the intuitive idea of discovering the maximal k-mutual-friend is to remove all the unsatisfied vertices and edges from G. It iteratively removes edges that are not contained in k triangles until all of them satisfy the condition that they have atleast k-mutual friends. When an edge is deleted, it decreases the triangle counts of the edges which are forming triangles with that edge. Thus we can obtain edges affected by the deleted edge and decrease triangle counts for them.

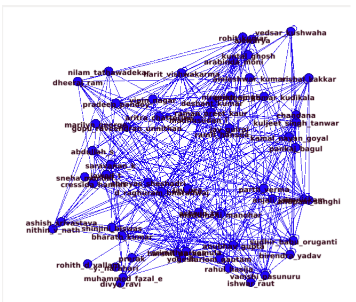
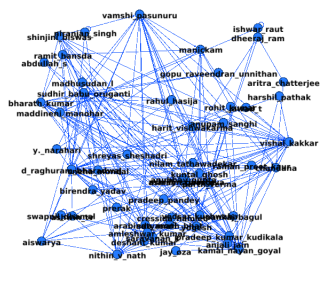
Complexity Analysis :

Step 1 : $O(|V|)$ to check vertices' degree.
Step 2 : The initial triangle counting has time complexity of $\sum_{v \in G} d(v)^2$.Finding all the edges forming triangles with the current edge $e(u, v)$ takes $d(u) + d(v)$ work. In the worst case, all the edges are removed from Q. Since Q only stores each edge one time, total cost is $\sum_{e(u,v) \in G}(d(u) + d(v))$, equal to $\sum_{v \in G} d(v)^2$.
Step 3 : $O(|V|)$ to remove isolated vertices.
Total time complexity of the algorithm :
The second step dominates the whole procedure. Hence the total time complexity is $O(\sum_{v \in G} d(v)^2)$.
Algorithm needs $O(|E|)$ space complexity.
M-KMF has high CC But low Modularity.

Experimental Results:

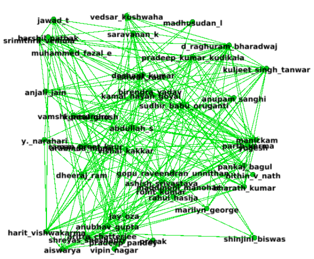
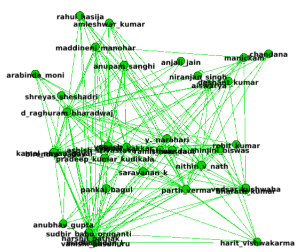
INPUT

Induced "Who Follows Whom" and Induced "Who Upvotes Whom" social network:

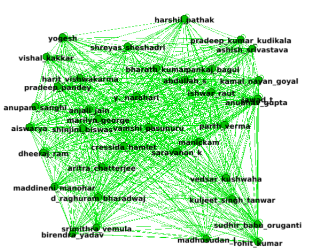
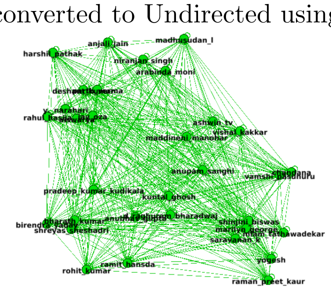



OUTPUT:

Here Directed Network is converted to Undirected using A +A'
In first network k=2 and in second network k=26.

Here Directed Network is converted to Undirected using Triadic Closure

Intuitively we can see that the manipulative community should have large cohesion within the community and less number of edges coming from outside.
NOTE: The course-instructor is coming as the part of the community which will not be the case in real life as the course-instructor will not be part of manipulative coalition. This is happening because the M-kmf is based on CC and only considers cohesion inside the community. So, we reject this measure because it may give undesirable results for our problem.
Next, we choose modularity as a measure.

6 Modularity Maximization [1]

In [6], Modularity Q of a graph is defined as follows :

$$Q = \sum_i e_{ii} - a_i^2$$

where e_{ii} is the fraction of edges that connects two nodes inside the community i and a_i represent the expected fraction of edges that connect two vertices in community i if the edges were distributed at random(i.e.having one or both vertices inside the community i).

Steps in the algorithm :

Step 1 : Finding edge weights

In this step we use labelling technique to find weights on the edges of the graph in a space-efficient manner. That is, in each iteration and for every vertex v, we assign all of its neighbour vertex u, a label v. Then, to find the number of common friends between v and each neighbour u of v, it is just enough to count all u's neighbour which have label v.

Edge weights are computed using the cosine similarity measure defined below :

$$S_{cosine}(i,j) = \frac{|N_i \cap N_j|}{\sqrt{|N_i||N_j|}}$$

$$\Delta Q = \frac{E_{ij}}{e} - 2\frac{2E_i + Ext_i}{2e}\frac{2E_j + Ext_j}{2e}$$

We select R(number of iterations of weighting algorithm) to be $log(n)$ where n is the number of vertices in the graph. In this situation, for each vertex the algorithm can find the weight of at least log(n) of its neighbour edges.

Step 2: Preliminary Community Detection

After weighting edges, we have to find preliminary communities (i.e. sub-communities). All weighted edges of the graph are first put in an array and then sorted based on their weight in descending order. At first, all vertices are considered unassigned. Then the edges in the array are picked one by one and for each edge an attempt is made to assign its vertices to a new community, if both of them were unassigned. That means, if we have an edge (u,v) that both of u and v have not already been assigned to a community, we create a new community and assign both vertices u and v to it . If at least one of them were already assigned to a community,we do nothing.

Step 3: Merging Communities

In "Hybrid" merging, we start with "pairwise" merging for some iterations. Having had R_m as the number of iterations for merging, we choose a fraction of it(i.e. Frac) to start with pairwise merging. The rest iterations(i.e. $(1 - Frac)R_m$) will be devoted to "single neighbours" merging. A reasonable value for R_m is log(n). Frac can be simply set to 0.5.

Pairwise merging : First, for each community c_i, we find a neighbour community c_j which merging them will result in maximum ΔQ. If this maximum ΔQ is positive then we draw an arrow from community c_i to community c_j. If there exists a bi-directional arrow between two communities c_i and c_j, merging this pair of communities will lead to more increase in modularity value in comparison with merging one of them with another single neighbour community. This kind of merging is referred to as "pairwise" merging.

Single neighbour merging : In this type of merging, all communities which are connected with only one community(with an arrow), will be merged to that corresponding community. Although it has more speed in merging, this approach will not guarantee to increase the modularity value. R_m is set to log(n).

Complexity Analysis :

Step 1 : Weighting Algorithm $O(R.m)$.

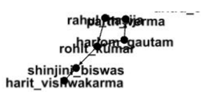
Step 2 : Preliminary community detection $O(mlogm)$.

Step 3 : Merging stage $O(mlogn)$.

The total time complexity of the algorithm is :
$R.D.n + D.n.log(n) + log(n).D.n$
where D = $\frac{\sum_{i=1} nd_i}{n}$, so D=$\frac{2m}{n}$, and $m = nD/2$.
Taking R = $\lceil log(n) \rceil$, the overall time complexity of the algorithm is $O(nlog(n))$.

Experiment Results:

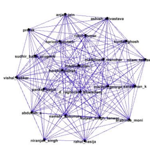
Top Communities based on the modularity value of the communities:
Here Directed Network is converted to Undirected using A +A'
Who follows Whom Network:

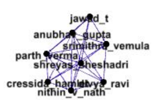


Q=0.055 Q=0.048
Who upvotes whom:

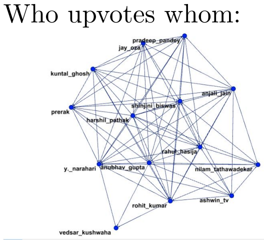
Q=0.07 Q=0.033

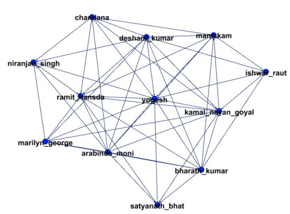
Here Directed Network is converted to Undirected using Triadic Closure
Who follows Whom Network:





Q=0.691 Q=0.081
Who upvotes whom:





Q=0.296 Q=0.1984
Modularity is giving very good result which is also conforming to real communities. And One more good thing is that the time complexity of algorithm is also low.
If the communities with high modularity also form cliques,then we can clearly deduce that such communities are manipulative.

7 Conclusion and Future Work:

Modularity works best for the problem of finding strategic manipulations in discussion forums.
We wish to define a penalty function so that the best strategy for every individual is not to form such coalitions.
The penalty imposed on members of a coalition may be proportional to the modularity of that coalition as modularity itself is indicative of the strength of such coalitions.

Higher the modularity, the higher will be the strength of community and which implies there should be more penalty to the member who is the part of coalition.

Acknowledgement

We would like to thank Prof.Y Narahari,Prof Siddharth Barman and our mentors Swapnil Dhamal and Amleshwar Kumar for their extensive advice and help in the project.

References

[1] Mohsen Arab, Mohsen Afsharchi, A Modularity Maximization Algorithm for Community Detection in Social Networks with Low Time Complexity,Conference: 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)

[2] Feng Zhao, Anthony K. H. Tung, Large Scale Cohesive Subgraphs Discovery for Social Network Visual Analysis, In Proceedings of the Very Large Databases(VLDB) Endowment, December 2012.

[3] Meng Wang, Chaokun Wang, Jeffrey Xu Yu, Jun Zhang, Community Detection in Social Networks: An In-depth Benchmarking Study with a Procedure-Oriented Framework, In Proceedings of the Very Large Databases(VLDB) Endowment, June 2015

[4] M. E. J. Newman, Modularity and community structure in networks. In Proceedings of the National Academy of Sciences of the United States of America 2006

[5] M. E. J. Newman, Detecting community structure in networks. The European Physical Journal B 2004

[6] M.Girvan , M. E. J. Newman , Community structure in Social and Biological Networks. Proceedings of the National Academy of Sciences of the United States of America 2002