Detection of Fraud Group in Online Communities by Clustering Algorithm

Mostafizur Rahman, Mohammad Mojahidul Islam

1. Abstract

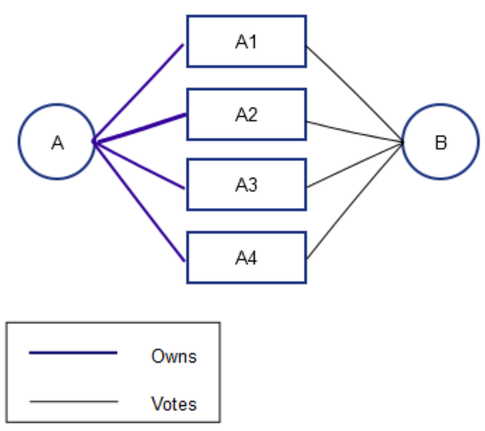
Online communities have become very popular in the recent times. In these communities people help each other by sharing knowledge, ideas, skills and even experiences. People primarily ask questions in the communities when they face any problem and others who have faced similar problem previously or know how to solve it answer to that question. Its important to make sure that the knowledge base of the online community is not biased or corrupted. In this article we have introduced the most common fraudulent activities found in the online communities and proposed a graph based clustering algorithm to detect the fraudulent groups.

2. Motivation

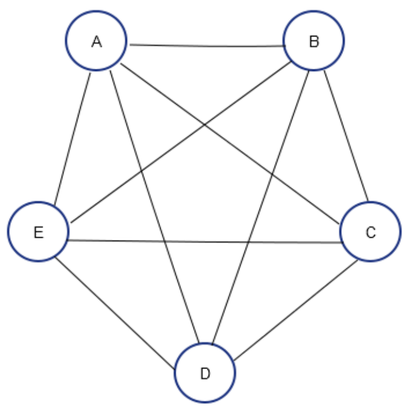
- Online communities have voting functionality to offer point/score if the answer is found useful by community users
- The more up-votes an answer has the more useful it is
- The points of the users has become a measure of their knowledge or skill
- Some users may form a group and vote for each other to gain more points making the knowledge base corrupted and the point system imbalanced
- This analysis is included in social network analysis which is an active research area

3. Problem Definition

- We have identified four key weakness in the current online community system:
- Serial up-voting/down-voting: Serial voting is to vote on many answers of a specific user in a short period of time
- Sock puppetting: Sock puppetting occurs when an user maintains a secondary account to vote on his answers to increase his point
- Voting group: Voting group is formed by some users when they votes each others' answers to increase their points



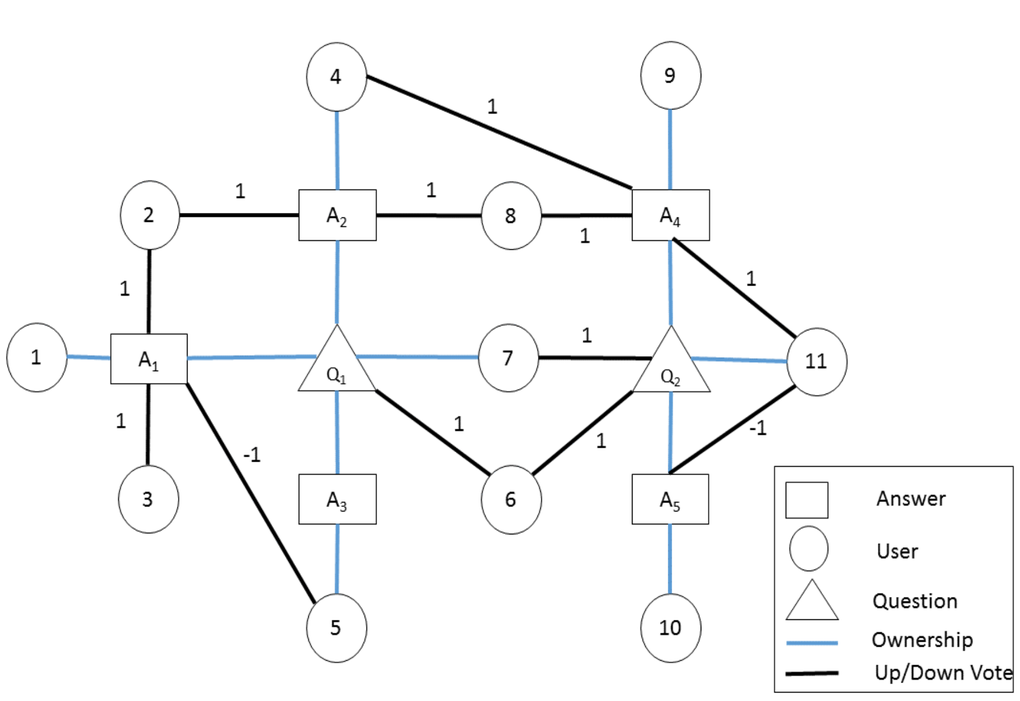
(a) Serial voting



(b) Voting group

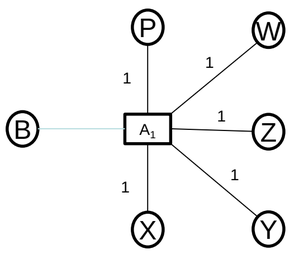
4. Attributed Graph Representation

- Three basic related entities: Question, Answers, User and their relations can be represented by an attributed graph (a graph containing structural and compositional information)
- Blue edges can happen between an user and a question or an answer. Positive (negative) black edges indicate up-vote (down-vote) points given by an user to a question or an answer

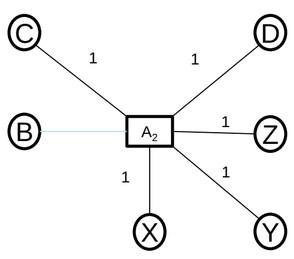


5. Superimposed Answer Graph

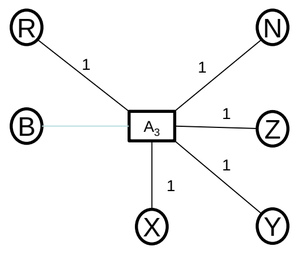
- Answer sub-graph of an answer is formed by taking the user nodes who up/down-voted the answer and their incident edges
- A superimposed answer graph of an user is the superposition of all nodes and edges of answer sub-graphs of all answers given by the user, i.e. taking union of nodes and edges. But weight of the edges are the sum of edge weights of answer sub-graphs



(a) Answer sub-graph of A_1

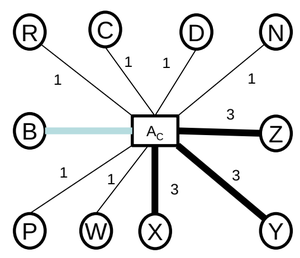


(b) Answer sub-graph of A_2



(c) Answer sub-graph of A_3

The B nodes and the blue edges are not part of any graph. They are shown to improvise understanding.



(d) Superimposed graph of B

6. Extension of Graph Modularity

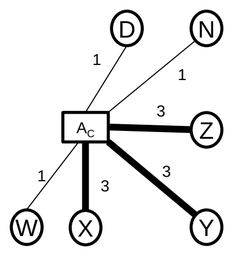
- Modularity is a measure of how much a graph is dense and modular resulting from a clustering of nodes into groups. Modularity is defined as

$$Q = \frac{1}{2m}\sum_{ij}(a_{ij} - \frac{k_i k_j}{2m})\delta(\gamma_i, \gamma_j)$$

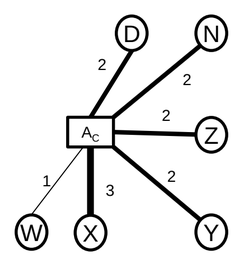
- This formula can adapted to our voting group detection. Modularity of a superimposed answer (node u) graph can be defined to indicate how much voting activity is confined to smaller number of users

$$Q(u) = \sum_{e \in E}\left(\frac{\omega(e)}{\sum_{e \in E}(\omega(e))} - \frac{1}{2m}\right)\frac{\omega(e)}{m}$$

- m = total number of edges, w(e) = weight of edge e. k_i , k_j is the degree of node i, j. In the figure, graph (a) has greater extended modularity than (b) since (a) has activity with smaller voting group (X, Y, Z). Kronecker delta between nodes i and j is 1 if they are in same cluster, otherwise 0



(a) Q = 0.25



(b) Q = 0.194

7. Proximity and Threshold Graph

- A similarity proximity measure can be defined between two user nodes (A, B) in terms of their extended modularity

$$P(A,B) = Q(A)\frac{\sum_{e \in SS}(\omega(e))}{\sum_{e \in S}(\omega(e))} + Q(B)\frac{\sum_{e \in RR}(\omega(e))}{\sum_{e \in R}(\omega(e))}$$

- First and second ratio term indicate contribution of B to A's probable fraud activity and contribution of A to B's fraud activity
- A threshold graph can be plotted to find maximal clique or connected components to detect fraud group. It contains all user nodes and edges between nodes if their P(A,B) ≥ a(threshold)

8. Results

- Clustering the user nodes to find fraud group is dictated by the choice of threshold (a)
- Poor choice of a finds groups that are not rigorously formed in practical scenario
- Very small choice of a adds more non-related user nodes to a real fraud group
- Very small high choice of a removes the related user nodes from the real fraud group.
- Detecting a fraud group also dictated by activity level. High activity in the network can generalize good result from the proposed technique
- Choice of algorithm has also effects on results.(i.e. Clique finding , Connected components)

9. Conclusion

- We assumed, high percentage of score sharing can lead to voting group. But in practical situation this can be co-incidental
- Domain based knowledge is not taken into consideration. Popularity measure can be introduced here since popular user always share a high percentage of score with its fans.
- Machine learning techniques can be adapted to the system to predict a good functional threshold value for threshold graph

Department of Computer Science and Engineering (CSE), BUET