Accurately Detecting Trolls in Slashdot Zoo via Decluttering

Padma Sai (2016A7PS0084P)
Purush(2016A7PS0025P)
Raghavendra(2016A7PS0107P)
Prof. Poonam Goyal
Information Retrieval
16 November 2018

ABSTRACT

Online social networks like Slashdot bring valuable information to millions of users - but their accuracy is based on the integrity of their user base. Unfortunately, there are many “trolls” on Slashdot who post misinformation and compromise system integrity. In this paper, we develop a general algorithm called TIA (short for Troll Identification Algorithm) to classify users of an online “signed” social network as malicious (e.g.trolls on Slashdot) or benign (i.e. normal honest users). Though applicable to many signed social networks, TIA has been tested on troll detection on Slashdot Zoo under a wide variety of parameter settings. Its running time is faster than many past algorithms and it is significantly more accurate than existing methods.

PROBLEM STATEMENT:

Online Social networks can represented as a weighted directed graph.It can Represented by G=(V,E,W),where V is a set of vertices,each vertex represents a user and an edge between users represents a relation between them(friend or foe).W:E --> [-1,+1] is a weight function that maps each edge with a real number between [-1,+1].(More on how to create these graphs from various social sites is mentioned below.)

TIA(Troll Detection algorithm) is a method to detect malicious users (or trolls in our case).It works in the following way - each user is assigned a score based on the centrality measure used and a centrality threshold.The algorithm works faster when a graph with relatively lower number of edges is given as a input.So in order to simplify a complex SSN(signed social network) into smaller and simpler SSNs several decluttering operations are introduced about which we will be discussing more elaborately later.A major challenge in the effective identification of the malicious users is that they take several carefully designed steps in order to evade the detection.In handling of these challenges these decluttering operations will prove to be effective.we show that under the appropriate conditions the TIA that is modified with decluttering operations has several advantages such as a) over 3 times the precision of the best existing algorithm to find trolls in Slashdot, (b) retrieves over twice the number of trolls that [1] does, and (c) does all this while running 25-50 times faster.

DESCRIPTION OF THE APPLICATION DOMAIN

The problem of detecting trolls in online environments like Slashdot and other signed social networks is increasingly important as open source, collaboratively edited information becomes used more widely. Ensuring the integrity of this information is important for users while posing a technical challenge as many entities have strong incentives to compromise the accuracy of such information.

MOTIVATION OF THE PROBLEM

Social media is a huge part of our lives. It's used in everything, and it's everywhere. We all use it; our friends use it, advertisers use it, celebrities, retail brands, etc. The world population is 7.3 billion and the Internet has 3.17 billion users. With these kinds of numbers, all types of users will undoubtedly be subjected to some form of social media trolling during their usage. Trolls, who create conflict on social media sites by making controversial statements with the purpose of causing havoc, can be found in almost every corner of the web. They can create problems for people, businesses, and the general well-being of some aspects of social media.Hence we chose this paper.

TECHNICAL ISSUES FACED

Initially we tried implementing without using any python library.we faced difficulty in implementing centrality measures like Signed Eigenvector Centrality(SEC) and Spectral eigenvector measure.And our code and data structures were not scalable.so we used networkx library (a python graph library).

The dataset is really huge approximately 140k edges .we had limited computing power to run our algorithm.

LITERATURE SURVEY

SIGNED SOCIAL NETWORKS(SSN): (referenced from : J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Predicting positive and negative links in online social networks, " in WWW, 2010, pp. 641-650.)

A Signed Social Network (SSN) is a directed, weighted graph G = (V,E,W) where V is a set of users, $E \subseteq V \times V$ is a set of edges, and $W : E \rightarrow [-1, +1]$ is a mapping. We can think of the mapping W(u, v) as a score of the amount of liking/friendship the user u has towards user v.

Derivation of SSNs from several social networking sites such as

Slashdot:

On the Slashdot we can define the SSN as W(u, v)=1,when user u explicitly marks user v as friend and W(u, v)=-1 when user u explicitly marks user v as foe.

Wikipedia:(referenced from:S. Maniu, B. Cautis, and T. Abdessalem, “Building a signed network from interactions in wikipedia,” in DBSocial, 2011, pp. 19–24)

On Wikipedia, we can define W(u, v) using NLP in many ways. One way is to set

$$W_{wi}(u,v) = \frac{\sup p(u,v) - rev(u,v)}{edits(v)}$$

where *edits* (v) is the set of all edits made by v during some fixed time window, *SUPP* (u,v)is the number of edited documents in *edits* (v) that were subsequently edited but not substantively reverted by v, and *rev* (u,v) is the number of edits in *edits* (v) that were subsequently reversed by v.

YouTube:

One way to derive an SSN from YouTube is to set:

$$W_{yt}(u,v) = \frac{thumbs\_up(u,v) - thumbs\_down(u,v)}{|posts(v)|}$$

where *posts* (v) is the set of all videos posted (during some fixed time frame) by v that were marked positively or negatively by u,thumbs-up(u,v) is the number of videos in *posts(v)* marked with a "thumbs up" and "thumbs_down" is the number of videos in *posts(v)* marked with a thumbs down.

Twitter/Facebook:(referenced form : R. Feldman, "Techniques and applications for sentiment analysis," Commun. ACM, vol. 56, no. 4, pp. 82–89, 2013)

Given an edge (u, v) denoting that u follows v on Twitter (or friends v on FB), we set:

$$W_{tw}(u,v) = \frac{pos(u,v) - neg(u,v)}{|tweets(v)|}$$

where tweets(v) is the set of tweets posted by v during a given time frame, pos(u, v) is the subset of those tweets that are either retweeted by u or essentially rephrased by u

afterwards, and neg(u, v) is the subset of tweets(v) that are the sets of tweets in pos(u, v) whose content is contradicted by a subsequent post by v.we have to note that NLP(Natural Language Processing) techniques that use sentiment analysis can be used to estimate the sets pos(u, v) and neg(u, v).

CENTRALITY MEASURES FOR SSNs:-

Given a SSN G=(V,E,W), a node centrality measure is a function R:V→R that assigns a score to each user-the higher the score, the more important the user i.e. a benign user will have higher score when compared to malicious users.various centrality measures have been used till now and now we will review them.

Freaks

The freak score of v is the number of incoming negative edges. For weighted networks, the *Freak Centrality* of v is the sum of the weights of incoming negative edges.

$$freaks(u) = \sum_{v \in V | W(v,u) < 0} W(v,u)$$

Fans Minus Freaks (FMF)

The centrality of u is the number of positive incoming edges (*fans*) minus the number of negative incoming edges (*freaks*). For weighted networks, we define FMF centrality as the total positive incoming weight minus the the total negative incoming weight.

$$FMF(u) = \sum_{,v \in V | W(v,u) > 0} |W(v,u)| - \sum_{v \in V | W(v,/u) < 0} |W(v,u)|$$

A similar measure, called ***Prestige*****,** has been proposed and is obtained by dividing the FMF centrality by the sum of the absolute values of the incoming weights for each node.

Page Rank (PR)

PageRank is defined for directed graphs with non-negative edge weights. It was originally developed for indexing web pages, and represents the likelihood that a person following links will arrive at a particular page. The PageRank of a node u is defined as

$$PR(u) = \frac{1-\delta}{|V|} + \delta \sum_{v \in pred(u)} \frac{PR(v)}{|succ(v)|}$$

Here, δ is a "damping factor" (usually 0.85) which captures the probability that a user arrives at a web page by following links (as opposed to landing on the page via some other process). pred(u) is the set of all vertices v such that $(v,u) \in E$ and *succ(v)* is the set of all vertices v' such that $(v,v') \in E$.

A ***Modified PageRank*** **(M-PR)** has been proposed to take into account both positive and negative links. In particular, they apply PageRank separately on A^+ (sub-network with positive links) obtaining PR^+, and on A^- (sub-network with negative links) obtaining PR^-. The final rank vector M-PR is computed as M-PR $= PR^+ - PR^-$.

Signed Spectral Ranking (SSR)

Signed Spectral Ranking (SSR) improves upon PageRank by taking edge signs into account. It is computed by taking the dominant left eigenvector of the signed matrix

$$G_S = \delta \cdot H_A + \frac{(1-\delta)}{|V|} J_{|V|\times|V|}$$

Positive edges correspond to endorsements, while negative edges to criticisms.

Negative Ranking (NR)

An empirical evaluation of SSR and PR using Slashdot data was done and it shows that SSR and PR values were almost equivalent for benign users, but PR value for trolls was much more than their SSR value. They suggest a Negative Rank measure computed by subtracting PR from SSR, i.e. $NR(u)=SSR(u)-\beta \cdot PR(u)$, where β is a parameter determining the influence of PageRank on the ranking. As empirical study obtained its best results when $\beta=1$, we use $\beta=1$.

Signed Eigenvector Centrality (SEC) (referenced form : P. Bonacich and P. Lloyd, "Calculating status with negative relations," Social Networks, vol. 26, no. 4, pp. 331 – 338, 2004)

Eigenvector centrality (EC) was proposed by Bonacich for networks with nonnegative edge weights given by the dominant eigenvector of the adjacency matrix. As eigenvectors can be computed for any matrix, a study on "Calculating status with negative relations, "suggests that this measure can also be computed for (weighted) signed networks. Thus, the signed eigenvector centrality of a vertex v can be computed from the vector x that satisfies the equation $Ax=\lambda x$, where λ is the greatest eigenvalue.

Modified HITS (M-HITS) The HITS link analysis algorithm to rate Web pages has been adapted for SSNs by iteratively computing the hub and authority scores separately on A^+ and A^-, using the equations:

$$\begin{cases} h^+(u) = \sum_{v \in succ^+(u)} a^+(v); \; a^+(u) = \sum_{v \in pred^+(u)} h^+(v) \\ h^-(u) = \sum_{v \in succ^-(u)} a^-(v); \; a^-(u) = \sum_{v \in pred^-(u)} h^-(v) \end{cases}$$

and by assigning, after convergence, the score $a(u) \simeq a^+(u) - a^-(u)$ to each node ′ul.$pred^+(u)$(resp. $pred^-(u)$) denotes the set of nodes v in *pred* (u) s. t. W(v,u)>o(resp. W(v,u)<o). Similarly for $succ^+(u)$ and $succ^-(u)$.

Bias and Deserve (BAD) a node v's bias (BIAS) reflects the expected weight of an outgoing connection, while its deserve (DES) reflects the expected weight of an incoming connection from an unbiased node. Similarly to HITS, BIAS and DES are iteratively computed as:

$$\begin{cases} h^+(u) = \sum_{v \in S'ucc+}(u)a^+(v), a^+(u) = \sum_{v \in pred+}(/\alpha)h^+(v) \\ h^-(u) = \sum_{v \in succ^-(,u)} a^-(v), a^-(u) = \sum_{v \in pred^-(u)} h^-(v) \end{cases}$$

where $X_t(v,u)=\max(o. BI \text{ Ast } (v)W(v,u))$. Finally, the scores of the nodes in the SSN are taken as the vector *DES*.

Requirements of a Good Scoring Measure

A set of axioms are proposed which suggest that a good measure of centrality in SSN s must satisfy under the assumption that a set of nodes is benign.

Axiom 1:-

A positive edge from a benign node to a node v *should increase* v *'s centrality*. Intuitively, a positive edge from a benign node to another node means that the benign node also thinks the other node is benign-otherwise there is no reason for the benign node to implicitly endorse the other node.

Axiom 2:

A negative edge from a benign node to a node v should decrease v's centrality. As in the previous axiom, benign nodes have an incentive to identify malicious nodes in order to preserve the integrity of the social network as a whole. As a consequence, when a benign node says a node is malicious, there is some chance that it actually is malicious.

Axiom 3:

A positive edge from a malicious node to a node v should decrease v's centrality. The rationale behind this axiom is that malicious nodes have a strong incentive to endorse other malicious nodes so that an "army" of malicious nodes can collectively perform some task(s).

Axiom 4:

A negative edge from a malicious node to a node v should increase v's centrality. As in the previous case, malicious nodes have an incentive to downgrade the centrality of benign nodes so that, in comparison, other malicious nodes get a high score. As a consequence, when a node is disliked by a malicious node, it probably means that the malicious node is trying to "bad mouth" a benign node.

Attack Models (referenced from:D. Donato, S. Leonardi, and M. Paniccia, "Combining transitive trust and negative opinions for better reputation management in social networks," in SNA KDD, 2008)

In the real world, benign users can be tricked into endorsing malicious users (via positive outgoing edges). For example, benign users may endorse someone because they were endorsed by that user, not necessarily because they like that user. Malicious users may

endorse a benign user (i.e. have positive edges to benign users) in the hope that the benign user reciprocates, which would increase their centrality. In such cases, past methods to identify malicious nodes in SSNs can lead to error as shown in Example 2 and the following discussion. Specific attacks described in [17] include:

(A) Individual malicious peers.

Malicious users always present bad behavior, and hence receive negative links from good users. These are relatively stupid malicious users who should not be difficult to detect, say by using Freaks centrality

(B) Malicious collectives.

Malicious users endorse other malicious users. In this case, a malicious user's score may increase due to the presence of a bunch of positive incoming links.

(C) Camouflage behind good transactions.

Malicious users can cheat some benign users to vote positively for them. This happens, for instance, when malicious users endorse a benign user who, out of courtesy, endorses them back.

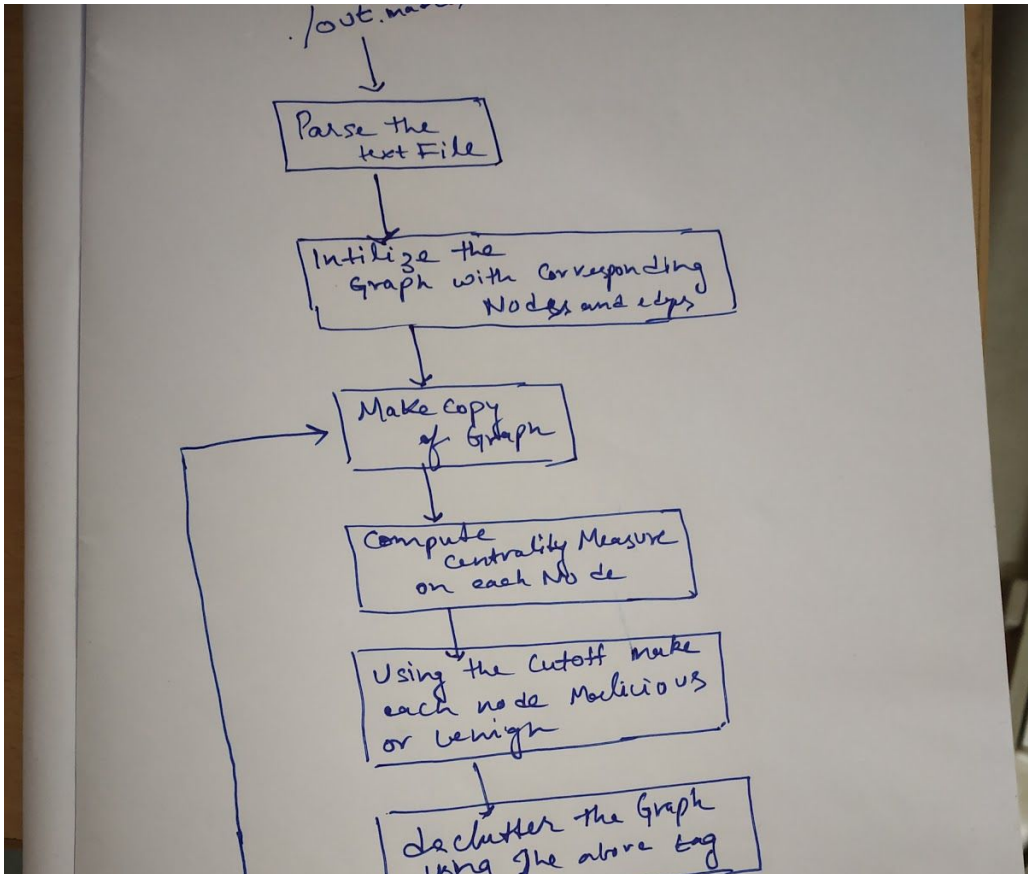
(D) Malicious spies.

There are two kinds of malicious users: some of them act as in threat models B and C, while the others (called spies) make benign users to vote positively for them, and assign positive value only to bad nodes.

(E) Camouflage behind judgements.

The strategy in this case is to assign negative value to good users. Then, this can cause the decrease of rank for good peers, and, consequently, the increase of malicious user's rank.

FLOWCHART


Parse the text File
Intilize the Graph with corresponding Nodes and edges
Make copy of Graph
Compute Centrality Measure on each Node
Using the Cutoff make each node Malicious or benign
declutter the Graph

RESEARCH GAP:

There is much future work to be done. we can combine the power of natural language processing methods and network analysis methods to find trolls. Clearly, looking at the content of posts on Twitter or Facebook would help find troll like individuals in Facebook or Twitter-on Wikipedia, finding vandals involves NLP as we need to understand the relationship between changes made by a user and the previous content in order to check whether the edits reverted or contradicted what had previously been said.

CONCLUSION

We successfully implemented the algorithm.We tested the algorithm on the dataset mentioned in the paper.