Stemming the flow of information in a social network

Balaji Vasan Srinivasan*, Akshay Kumar*†, Shubham Gupta*†, Khushi Gupta*‡

*Adobe Research India Labs, Bangalore, India
‡Adobe Systems India Pvt. Ltd
†Indian Institute of Technology, Kanpur, India
balsrini@adobe.com,kakshay@iitk.ac.in,shubhamg@iitk.ac.in,khgupta@adobe.com

Abstract. Social media has changed the way people interact with each other and has contributed greatly towards bringing people together. It has become an ideal platform for people to share their opinions. However, due to the volatility of social networks, a negative campaign or a rumor can go viral resulting in severe impact to the community. In this paper, we aim to solve this problem of stemming the flow of a negative campaign in a network by observing only parts of the network. Given a negative campaign and information about the status of its spread through a few candidate nodes, our algorithm estimates the information flow in the network and based on this estimated flow, finds a set of nodes which would be instrumental in stemming the information flow. The proposed algorithm is tested on real-world networks and its effectiveness is compared against other existing works.

Keywords: social network, rumor source, rumor stemming, targeted influence

1 Introduction

Social media is changing the way people communicate with each other and have become the preferred mediums of communication between friends and family. These sites serve as excellent platforms for sharing information and is being extensively used /leveraged for several business applications such as advertising, marketing, e-business and social campaigning. The structure of these networks is such that there is huge potential for a particular piece of information to go "viral". Marketers hope to exploit this and use it to their advantage when designing advertisements /campaigns. However, this also allows the possibility of a negative or inauthentic piece of information to become widely popular. Such pieces of incorrect or malicious information against individuals or brands by various agents, adversaries or competitors can have severe detrimental effects tarnishing hard built brand images.

There are several cases of documented social media disasters - both at the enterprise level as well at the political level. Rumors about the outbreak of swine flu resulted in a large online panic in 2009[4]. GreenPeace's extensive campaign against Nestle's misuse of the Indonesian rainforests lead to serious damage of the Swiss based corporation's brand image[17]. The ethical implications in these examples (attempts to handle inconvenient truths, etc.) are not addressed here and are beyond the scope of this paper. However, these examples also warrant for a method to limit the effect of an information deemed as negative, which will be the primary subject of our work.

There are two stages to stem the spread of a misinformation. The spread needs to be estimated first, followed by approaches to limit it. The former problem has been studied extensively in the lights of disease spread [2, 12, 14]. The latter problem has been recently studied as the "Influence Limitation" problem[1, 13]. In this paper, while we focus on the overall problem, our primary innovation is around the latter part of the problem.

Existing work in influence limitation removes the affected part of the network and maximizes the influence in the unaffected to limit the overall influence spread. However, real-world networks can seldom be differentiated into these 2 clear classes. To address this, we propose a methodology to score each users based on their "affectedness" to the negative campaign. We start with an estimate of the spread of negative information across the whole network and then determine the vulnerability of the nodes to the negative campaign to categorize them as *infected* (the negative information has already reached these nodes, the nodes are affected by the negative information and have started spreading them), *vulnerable* (the negative information may or may not have reached these nodes, but are most likely to move to the *infected* state in the near future) and *un-infected* (immune to the spread of the negative campaign) categories. Limiting the spread will then amount to targeting a "positive" campaign to the users in the vulnerable category (and may be the uninfected as well).

This paper is organized as follows. In Section 2, we discuss related prior work, highlighting their shortcomings and how we will overcome them. Section 3 provides an overview of our end-to-end approach for stemming the flow of negative information followed by the discussions of our algorithms to rank order the nodes based on their "vulnerability" in Section 4. We present the performances of our algorithm in Section 5 compared against existing work. Section 6 concludes the paper.

2 Related Work

The first stage of the problem is to estimate the spread of the information in a social network and could be cast as the "Rumor Source Identification" problem. Shah and Zaman [15] developed rumor-centrality to identify single-source in the epidemic. Prakash et al. [12] extended the source identification to address multiple sources via the use of Minimum Description Length and also developed mechanisms to identify the number of sources automatically. However, both these approaches require the complete snapshot of the spread in the network. To alleviate this issue, Seo et al [14] identify a few monitor nodes in the network and categorize them as positive and negative monitors on the basis of whether they have received the information or not. The rumor source identification is based on the intuition that the source should be close to positive monitors but far from the negative monitors. This approach is more practical because it requires the knowledge of the status of monitor nodes only unlike the entire network, although this may not be as accurate as the former approaches. Leskovic et al. [8] study this further by analyzing the placement of these monitor nodes to quickly detect rumor spreads.

The second stage of the problem requires identifying mechanisms to spread the positive word to the network. This is studied as the problem of Influence Limitation. This is a special case of the Influence Maximization problem, which was first studied by Domingos and Richarson [5] and formalized by Kempe et al [7]. Influence maximization has since been extensively studied (e.g. [3, 10]). Influence limitation in the context

of social networks is comparatively lesser studied despite having a close resemblance to the disease spread models [2, 12]. The problem of influence limitation in the presence of competing campaigns was first addressed by Budak et al [1]. Budak et al. show that this problem was NP-hard and in general does not follow the sub-modular property. They solved restricted versions of the problem that were proved to be submodular and provided approximation guarantees for greedy solutions. They compared the performance of their greedy algorithm against various heuristics like degree centrality. Premm et al [13] extended the work in [1] to solve the general influence limitation problem when the underlying functions are non submodular. They used the concept of Shapley Value from co-operative game theory in order to capture the marginal contribution of a node as the number of nodes it influences in the good campaign. They first calculate the effect of the negative campaign and remove the nodes affected after a delay r. The nodes with the highest Shapley Value among the remaining nodes are selected to seed the positive information. Tong et al. [18] take an alternative approach to influence limitation by effecting edge addition and deletion to limit the spread. This work is interesting, but will not work in an online social network where formation and deletion of edges are non-trivial. We take a similar but more feasible approach towards influence limitation. We first identify a set of nodes, who if inoculated, would ensure that the rest of the network does not hear about the information and target them with the positive information to limit the spread of the rumor.

3 Our Approach - Overview

The primary objective of our algorithm is to identify nodes vulnerable to a negative campaign that is already spreading in the network and target them with positive information to stem the flow/spread of the negative information.

To estimate the spread, we follow the approach in [14] and assume that the status of a few nodes in the network can be accurately known. These nodes known as the **monitors** could be a set of brand loyalists, evangelists or well-wishers in case of an enterprise who can accurately report about the negative campaign. Monitors who have received the information are termed as positive monitors and those who have not are negative monitors. We use the approach in [14] to estimate the source of rumor and use standard diffusion models to estimate the resulting spread [7].

Once the network infection status is approximately known, we score each node in the network in the order of their susceptibility to the negative information. We categorize the nodes into 3 types - *infected*, *vulnerable* and *uninfected* which are defined as below:

[INFECTED] Nodes that have seen the rumor and have started spreading it - these nodes are difficult to cure. We can neither influence them with our positive campaign nor use them for our positive campaign since they are sufficiently convinced about the negative campaign to spread them. It is difficult to cure these infected nodes with the positive information.

[VULNERABLE] Nodes that have either seen the rumor and are not spreading it or nodes that may see the rumor "soon" from one of the infected nodes. This set of nodes is critical to stemming the flow of rumor, because if not attended to, they may quickly get infected and start spreading the rumors. Therefore, these nodes must be reached out with the positive information.

[UNINFECTED] Nodes that are not infected and may not be infected soon. These are users not in the vicinity of the spread and can be ignored.

We identify the vulnerable nodes via the scores from two algorithms (described in Section 4 to compute the susceptibility score of the network. After identifying the "highly susceptible" targets, we identify influencers who can reach out to them efficiently. To effectively target the nodes we use the approach in [16] that extends the Maximum Influence Arborescence [3] to maximize influence on a specific set of targets. The complete flow of the proposed framework is shown in Figure 1.

Fig. 1. [color] *Flow process of the algorithm*

4 Vulnerability of nodes

We categorize the nodes by calculating their **vulnerability** using 2 different algorithms: Algorithm to measure reachability of node from the estimated sources; and Algorithm to measure susceptibility of nodes who have not seen the negative campaign

4.1 Reachability of nodes

This algorithm is based on the premise that the impact of a certain piece of information depends not only strength of the connection *via* which information is pervading but also on the relative prominence of nodes in a network. Our hypothesis here is that information is likely to propagate faster through an edge if it is from a prominent source than if it is from a less prominent source. We therefore consider two factors:

Information Diffusion: Probability of information diffusion from one node to another is generally captured as the weight of the edge between them. In our algorithm, we use the approach in [16] to determine these influence probabilities.

Importance: Relative importance of the users along the path of information transfer is measured using their page rank [11] in the entire graph. Page rank is a popular measure of centrality of the nodes in a network. A higher value of page rank indicates that the node is better connected and hence more prominent in the network.

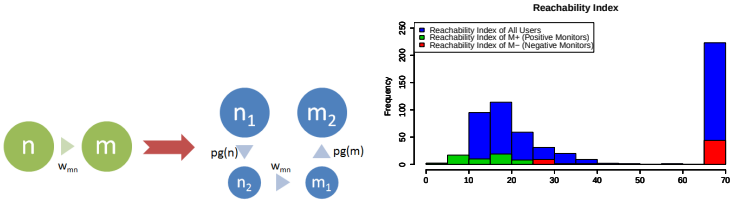
Let Σ denote the set of paths between the rumor source R and a node n. Consider one such path $\sigma \in \Sigma$. Suppose the path is as follows: $R \rightarrow i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{n-1} \rightarrow i_n$. We define a reachability index RI to this path σ as the product of the diffusion probabilities in the entire path and the importance of the nodes along the path:

$$RI(\sigma) = w(R, i_1) \left(\prod_{j=1}^{k} pg(i_j) w(i_j, i_{j+1}) \right) \tag{1}$$

Here, $w(i, j)$ is the weight on the edge based on the diffusion probability between nodes $i\&j$ and $pg(i)$ is the page rank of the node i. The idea is to account for the path of information flow based not only on the diffusion probabilities but also the prominence of the nodes along the path. The reachability index associated with this path is thus the product of probabilities of influence along the edges and page ranks of the nodes encountered on this path. The reachability index associated with all the paths between a pair of nodes are computed and the highest reachability index is chosen.

However, calculating all the paths between two nodes in a graph is NP-hard. To scale this scoring, we utilize the following modification:

1. The weight on the edges w_{ij} are modified as the $-\log w_{ij}$.
2. Every node n of the original graph G is converted into a directed edge $n_1 \rightarrow n_2$ in a new graph G' with a weight as $-\log pg(n)$. All the incoming edges into node n would now go into n_1 whereas all the outgoing edges from n would now emanate from n_2. In effect, n_1 has only incoming edges barring the outgoing edge $n_1 \rightarrow n_2$ edge and n_2 has all outgoing edges except for the incoming edge $n_1 \rightarrow n_2$ edge.
3. The transformation of an edge $n \rightarrow m$ is shown in Fig. 2(a).



(a) Graph Transformation (b) Histogram of Reachability Index

Fig. 2. Fig. 2(a): *Transformation of the graph edges to account for page rank.* ; **Fig. 2(b):** *Histogram showing reachability index of nodes. The ones at the extreme end of the histogram are those nodes which are not connected to rumor source.*

With these modifications, we can apply Dijkstra's Algorithm or any other shortest path finding algorithm to find the shortest path between the start node R and a node n to compute the susceptibility index of the node n. This algorithm is summarized in Algorithm 1. The $REACH(n)$ is defined as,

$$REACH(n) = \exp -RI(i) \tag{2}$$

The exponential scaling is done to negate the logarithmic scaling in our modification for Djikstra's algorithm.

Figure 2(b) shows the histogram for $REACH(i)$ *i.e.* $e^{-RI(i)}$ where i is a node in the graph for a sample graph with a randomly chosen rumor source. All the positive monitors have a higher susceptibility than the negative monitors, $\min_{i \in M^+} REACH(i) > \max_{i \in M^-} REACH(i)$, where M^+ indicates the set of positive monitors and M^- indicates the set of negative monitors. This is used to define a threshold for node categorization into infected or uninfected. The threshold θ_1 is defined as the geometric mean between the maximum of positive monitors and minimum of negative monitors, given by $\theta_1 = \sqrt{\left(\min_{i \in M^+} REACH(i)\right)\left(\max_{i \in M^-} REACH(i)\right)}$. Nodes having REACH greater then θ_1 are termed as *uninfected* whereas the ones having reachability index less then θ_1 are further classified as *infected* and *vulnerable*. The second level categorization is obtained by defining a boundary θ_2 at $1 - \sigma$ (standard deviation) from the mean of the reachability index score of all nodes. Nodes with reachability index between θ_1 and θ_2 are categorized as *vulnerable* and rest as *infected*.

4.2 Susceptibility of the node

The objective of this algorithm is to determine the susceptibility of a node based on its neighborhood. The hypothesis here is that vulnerable nodes have a heavily *affected*

Algorithm 1 Algorithm to measure reachability of node from the estimated sources

Require: $G = (V, E, w), R$ where w is the probability of influence along the edges and R is the rumor source

$G' = (V', E', w')$: new directed weighted graph
for i in V **do**
  Add i_1 and i_2 to V'
  Add (i_1, i_2) to E'
  $w'_{i_1,i_2} = -\log pg(i)$ where $pg(i)$ is the pagerank of node i in G
end for
for (i, j) in E **do**
  Add (i_2, j_1) to E'
  $w'_{i_2,j_1} = -\log w_{i,j}$
end for
for i in V **do**
  if R and i_1 are connected in G' **then**
    $RI(i) = shortest\_path\_length(G', R, i_1)$
  else
    $RI(i) = \infty$
  end if
end for
for i in V **do**
  $REACH(i) = e^{-RI(i)}$
end for
return REACH

neighborhood. However, certain nodes that do not have a heavily affected neighborhood need to considered as well if they have the potential to affect a large part of unaffected nodes when and if they get affected. To encapsulate these conditions, we consider the following factors for the susceptibility score:

Susceptibility: We calculate the susceptibility using the predicted snapshot of the network at time r. For the nodes which will get infected at time $r+1$, we look at all the incoming edges. The susceptibility is calculated as the sum of those edge weights which come from *affected* nodes, $sus(v)^r = \sum_{(u,v)\in E \wedge Affect(u)=1} w(e_{uv})$ We add a decay factor to the susceptibility to account for the recency in the infected neighborhood, $sus(v)^{r+k} = sus(v)^r e^{-\mu(k)}$ where μ is the decay factor.

Contagiousness: For each node that has not been affected at time r we calculate how contagiousness the node is by calculating the fraction of shortest paths from affected to unaffected nodes that pass through the node. This is a modified version of the popular *between-ness centrality* $bc(v) = \sum_{\substack{\forall(s,t)\in E \\ s\in Affected \\ t\in Unaffected}} \frac{\sigma_{st}(v)}{\sigma_{st}}$ where $\sigma_{s,t}(v)$ is the number of shortest paths from node s to node t that pass through v and $\sigma_{s,t}$ is the total number of shortest paths from node s to node t. This measure is akin to percolation centrality introduced in [6]. This measures how critical is the current node to the percolation of information from the infected to the uninfected parts of the network. A node with high contagiousness though not immediately vulnerable must be protected in order to prevent serious information spread.

Algorithm 2 Vulnerability Score

Require: $G = (V, E, w)$
1: $G' = (V', E, w)$: Status of Graph G after delay r
2: **for** k in $1 \rightarrow \infty$ **do**
3: $G' = (V', E, w)$: Status of Graph G after time $r + k$
4: **for** $i \in V$ *and* i *is not* $infected$ **do**
5: $sus(i) = \sum_{(u,v) \in E \ \wedge \ Affect(u)=1} w(e_{uv}) e^{-\mu(k)}$
6: $bc(i) = \sum_{\substack{\forall (s,t) \in E \\ s \in Affected \\ t \in Unaffected}} \frac{\sigma_{st}(v)}{\sigma_{st}}$
7: **end for**
8: **end for**
9: **return** $sus + bc$

The final susceptibility score is the sum of the susceptibility at r and the contagiousness of a node. The algorithm is briefed in Algorithm 2. The nodes that are not connected to any infected nodes are not infected at all and belong to Category 3 of nodes. Among the rest, the categorization is done by choosing the top $50-$percentile as infected and the rest as vulnerable.

4.3 Reachability & Susceptibility

Reachability score considers the position of a node with respect to the affected parts and the path to the affected part of the network to determine its vulnerability. Susceptibility score on the other hand accounts for the node's presence in the paths from the affected parts of the network. Both the approaches account for the factor of delay after which the information has reached a particular node. In the reachability, this is implicit in the product of weights along the path whereas the susceptibility score has an explicit exponential time decay factor.

One difference between the two Algorithms (Algorithm 1 and 2) is that Algorithm 1 finds targets who may or may not have seen the rumor/negative campaign and are vulnerable, whereas Algorithm 2 finds the scores based on connectedness to already infected nodes. However, both the algorithms can be used to identify vulnerable nodes that have to be targeted with the positive campaign.

Reachability is loosely related to the popular Independent Cascade Model (ICM) since it considers the independent paths between the nodes. Susceptibility on the other hand is loosely related to the Linear Threshold Model (LTM) since it evaluates the simultaneously affected neighborhood of a node.

We put nodes categorized as vulnerable by either of the algorithm into the *vulnerable* set. From the remaining nodes, those that are classified as infected by either of the approaches are put into *infected* set and the rest form the uninfected part.

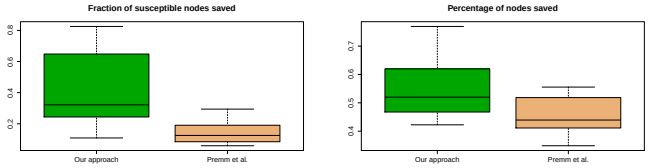
5 Experiments

All our experiments were performed on a data set collected by querying for the followers of the twitter handle `@barcampbangalore`. For each follower, we obtain the information about their connections and tweets using standard Twitter APIs. The final graph is built from the connections between all the followers. An edge is defined between every node and its set of followers. The edges are weighted using the approach

in [16]. The network had 571 nodes and 5747 edges. We use the rumor source identification in [14] with randomly chosen monitors in all our experiments. We used 25 ($\tilde{5}\%$ of the network size) monitors for our experiments.

Each of our experiment is repeated across 100 trials by randomly selecting a source and simulating the information flow based on the edge weights and considering this as our ground truth. Information flow is simulated using a Independent Cascade Model (ICM) [7]. Since the ground truth is generated via simulation, we present our results with this dataset only, however, we have observed that the performance holds for other networks as well.

We determine the nodes using our approach in Section 4.3 and target them by identifying the influencers using the algorithm in [16]. We compare our performance with the approach in [13] and compare the spread of the positive information using the Multi-Campaign Independent Cascade Model (MCICM)[1].



(a) Fraction of nodes saved (b) Percentage of nodes saved

Fig. 3. [color] *Performance comparison across configurations*

In our first experiment, we calculate the fraction of network saved by our algorithm. We estimate the spread the rumor without any positive campaign in the network using the standard ICM and then compare it with the case had there been a positive campaign propagating using the MCICM model [1] and calculate the number of nodes saved from the infection (nodes that received the information in the former setup and did not receive the information in the latter). Figure 3(a) shows the fraction of susceptible nodes saved in each of the configuration. Our approach performs significantly better than the Shapely value based approach in [13] in best/worst/average cases as evidenced by the box plot. The Mann-Whitney statistic [9] for the results from the two approaches had a p-value of 0.000367 (less than 0.05) indicating that they are statistically significant and the performance is not by chance.

In our next experiment,we find the percentage reduction in the infected nodes via the use of MCICM [1]. The resulting performance is shown in Fig 3(b), where again our approach performed better than the existing model from [13] in most cases. The Mann-Whitney statistic [9] for this experiment was observed to be 0.00235 again indicating statistical significance.

6 Conclusion

In this work, we have devised a mechanism for containing the spread of rumor in a network. The problem consists of two stages, identifying the spread of the rumor and then limiting it. Existing work solve only one part of the issue. However, we have proposed a comprehensive framework and also a new algorithm to score users vulnerable to the spread. We have compared our work with existing approach in [13], and the results are encouraging.

References

1. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the spread of misinformation in social networks. In: Proceedings of the 20th international conference on World wide web. pp. 665–674. ACM (2011)
2. Capasso, V., Capasso, V.: Mathematical structures of epidemic systems, vol. 88. Springer (1993)
3. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1029–1038. ACM (2010)
4. Chew, C., Eysenbach, G.: Pandemics in the age of twitter: content analysis of tweets during the 2009 H1N1 outbreak. PloS one 5(11), e14118 (2010)
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 57–66. ACM (2001)
6. Gönci, B., Németh, V., Balogh, E., Szabó, B., Dénes, Á., Környei, Z., Vicsek, T.: Viral epidemics in a cell culture: novel high resolution data and their interpretation by a percolation theory based model. PloS one 5(12), e15571 (2010)
7. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 137–146. ACM (2003)
8. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 420–429. ACM (2007)
9. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)
10. Narayanam, R., Narahari, Y.: A shapley value-based approach to discover influential nodes in social networks. IEEE Transactions on Automation Science and Engineering (99), 1–18 (2010)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. (1999)
12. Prakash, B.A., Vreeken, J., Faloutsos, C.: Spotting culprits in epidemics: How many and which ones? In: ICDM. vol. 12, pp. 11–20 (2012)
13. Premm Raj, H., Narahari, Y.: Influence limitation in multi-campaign social networks: A shapley value based approach (2012)
14. Seo, E., Mohapatra, P., Abdelzaher, T.: Identifying rumors and their sources in social networks. In: SPIE Defense, Security, and Sensing. pp. 83891I–83891I. International Society for Optics and Photonics (2012)
15. Shah, D., Zaman, T.: Rumors in a network: Who's the culprit? Information Theory, IEEE Transactions on 57(8), 5163–5181 (2011)
16. Srinivasan, B., Anandhavelu, N., Dalal, A., Yenugula, M., Srikanthan, P., Layek, A.: Topic-based targeted influence maximization. In: Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on. pp. 1–6 (Jan 2014)
17. Steel, E.: Nestlé takes a beating on social-media sites. The Wall Street Journal 29 (2010)
18. Tong, H., Prakash, B.A., Eliassi-Rad, T., Faloutsos, M., Faloutsos, C.: Gelling, and melting, large graphs by edge manipulation. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 245–254. ACM (2012)