How to Win Elections

Abdallah Sobehy(✉), Walid Ben-Ameur, Hossam Afifi, and Amira Bradai

Samovar, CNRS, Télécom SudParis, University Paris-Saclay,
9 Rue Charles Fourier, 91000 Évry, France
a.sobehy@gmail.com, walid.benameur@telecom-sudparis.eu

Abstract. Consider an election with two competing candidates and a set of voters whose opinions change over time. We study the best strategies that can be used by each candidate to influence voters. We also evaluate the knowledge advantage when one of the candidates knows in advance the adversary's strategy. We prove that an economy of up to 50% of the budget can be saved in such a scenario.

Keywords: Opinion dynamics · Reputation systems · Trust management · Social networks · Random graphs · Mixed integer programming

1 Introduction

Consider a social network where members interact within their social neighborhood leading to some opinion dynamics. Opinions can change when some kind of influence is applied on the members. Assume that two candidates are competing to win an election. The aim of this paper is to compare strategies that can be used to win an election given a limited budget to influence voters.

Elections should be understood in a broad sense since we might be interested in the propagation of two opposite pieces of information through a social network. In a distributed system, we might need to evaluate the authenticity of some information [2], the popularity of some contents [3–5] based on the contribution and collaboration of distributed agents. In network security context [6], a node whose neighbors are attacked becomes vulnerable leading to vulnerability propagation. Countermeasures are implemented to stop this propagation.

Another example of opinion dynamics is related to reputation of agents. As per the taxonomy defined in [1], reputation systems are divided into two main categories: implicit and explicit. Explicit reputation systems have an underlying implementation to assist members in assessing the trustworthiness of some entities. Examples of such systems are: Stackoverflow [7] where members who ask/answer good quality questions/answers are given reputation points; EigenTrust [8] which is used in P2P networks to choose trusted members to download files from; or more robust approaches as proposed in [9]. While in implicit reputation systems, there is no structured implementation for managing reputation. However, members use available reputation related information in decision making. Voting systems and social networks such as Twitter [10], and

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 221–230, 2017.
DOI: 10.1007/978-3-319-59288-6_20

Facebook [11] are examples of implicit reputation systems. Modeling such systems can help predict reputable members and the effect of their publications on others. Furthermore, it can be used to enhance explicit reputation systems as in SocialLink [12].

Regardless of the problem we are evaluating (popularity, reputation, authenticity, etc.), results depend on the underlying structure of the distributed system along with how agents update their opinions/beliefs. One general approach is based on imitation where each agent adopts the behavior or the opinion of some neighbors with some probability. An example of imitation is the DeGroot model [13] where each agent is updating his opinion using a weighted convex combination of neighbors' opinions. More sophisticated models are proposed in [9,14] allowing an agent to pay more attention to beliefs that do not differ too much form his/her own current opinion. Several other models are reviewed in [15].

For simplicity, we will only consider the DeGroot model in this paper. As previously mentioned, we assume that two candidates are competing to win an election. The set of voters will be modeled by a graph. Three types of graphs are considered in this paper and several strategies to influence voters are compared. We will also consider the case where one of the candidates knows exactly what will be done by the other candidate and tries to exploit this knowledge.

The organization of the paper is as follows: in Sect. 2 we introduce a Gossip-based model with stubborn agents. In Sect. 3 we consider this model in the context of elections where stubborn agents are the two candidates. Several strategies are then compared. A strategy based on prior knowledge of opponent's connections in order to win with a minimum budget is presented in Sect. 4. Conclusion and further research directions follow in Sect. 5.

2 A Gossip Model with Stubborn Agents

2.1 Overview

We propose an abstract solution to suit diverse contexts. Members of a network are represented as nodes of a graph. Each member has an opinion which describes his/her evaluation of a position. The value of the opinion is simulated as a continuous range, in our case from -1 to 1. This range can be a notion of trust towards a member where -1 means completely not trusted and 1 means completely trusted and 0 is neutral. Opinions disseminate through the network via edges with positive weight simulating the effect connected nodes have on one another; larger weight yields higher effect. Biased opinions are introduced by two forceful peers with unchanging opinions 1 and -1. Other peers are called normal peers with initial opinion of 0 that change over time. Forceful peers have been called stubborn agents in [16,17]. Each forceful peer tries to connect through the network in an efficient manner to have the majority of followers. A normal peer is considered as a follower of the positive opinion forceful peer if its opinion is greater than zero and vice versa. Peers whose final opinions are very close to 0 are considered neutral. Let us now describe the system model in details.

2.2 System Model

As previously mentioned, the network is represented by a graph; we will consider three types of graphs: Erdos-Renyi [18], Geometric [19], and Scale-free [20]. The graph is made up of N normal nodes and 2 forceful peers with different strategies which will be discussed later in details. The main target of our problem is to compute the opinions of nodes given the initial state of the graph topology and forceful peers. Let's assume the opinion of nodes are stored in an N+2 vector R where the first N elements R_N are the opinions of normal peers and the last two elements R_F are 1 and -1 respectively (opinions of forceful peers). Let R^i denote the final opinion of i^{th} node while O^i represents its initial opinion. Let $w_{i,j}$ be the edge weight between i^{th} and j^{th} nodes. The list of neighbors of the i^{th} node is noted neigh(i) while $w(i)$ denotes the sum of weights of incident edges $\sum_{j \in neigh(i)} w_{ij}$. We also use α to represent the weight given to the initial self opinion. *conv_thresh* is a convergence threshold used to decide whether convergence is reached. A small positive number *neut_thresh* is also used to define neutrality (if the opinion of an agent is between $-$ *neut_thresh* and *neut_thresh*, the agent is considered to be neutral). We assume that each normal peer is updating his opinion according to

$$R^i \leftarrow \alpha.O^i + \frac{1-\alpha}{w(i)} \sum_{j \in neigh(i)} R^j.w_{i,j}. \tag{1}$$

Put differently, each normal peer is combining his own initial opinion with the current opinion of the neighborhood. Writing this in matrix form, we get

$$R_N \leftarrow W_N R_N + W_F R_F + \alpha O_N. \tag{2}$$

where W_N is a $N \times N$ weight matrix, W_F is a $N \times 2$ matrix and O_N is a vector of size N representing the initial opinion of all normal peers. Matrix W_N is defined by $W_{Nij} = (1-\alpha)\frac{w_{i,j}}{w(i)}$ if $j \in neigh(i)$ and 0 otherwise. Observe that W_N is a substochastic matrix where the sum of elements of each row is strictly less than 1. This obviously implies that its spectral radius is strictly less than 1. Consequently, the sum $I + W_N + W_N^2 + W_N^3 + ...$ is an invertible matrix and its inverse is $I - W_N$, where I is the identity matrix. Observe also that by applying (2) k times, the obtained vector R_N is given by $W_N^k O_N + (W_N^{k-1} + ... + W_N + I)$ $(W_F R_F + \alpha O_N)$. This immediately proves that by repetition of (2), the vector R_N will converge and the limit is given by

$$R_N \rightarrow (I - W_N)^{-1}(W_F R_F + \alpha O_N). \tag{3}$$

Notice that the system does not generally reach a consensus. The limit depends on the internal structure of the graph (W_N), the initial opinion (O_N), the opinion of forceful peers (R_F), and how they are influencing the normal peers (W_F).

To compute the final opinions, we can either compute the inverse of $(I - W_N)$ or repeatedly apply (2). The latter way is illustrated in Algorithm 1. As stated above *conv_threshold* acts as the stopping condition for the algorithm.

Algorithm 1. R_N iterative update

$R^i = O^i$ **for** each $i \in N$
do
  $c\_op = R^i$ **for** each $i \in N$
  for each node $i \in N$ **do**
    $R^i = \alpha.O^i + \frac{1-\alpha}{w(i)} \sum_{j \in neigh(i)} c\_op^j.w_{i,j}$
  end for
  Update max_diff from R and c_op
while $\max_{i \in N} |R^i - c\_op| > conv\_threshold$

3 Strategies to Win an Election

We introduce 4 strategies by which forceful peers establish edges with normal peers given a bounded budget of connections (total edge weights for each forceful peer). Then we run Algorithm 1 to compute the final opinion for each normal peer. The forceful peer with the majority of followers is the winner. The simulation is run for different graph topologies and different budget values. We conclude this section with ordering strategies' efficiency for different graph configurations and formulating the relationship between followers' and winning percentages. For each strategy, forceful peers probabilistically choose their neighbors. 4 strategies are considered depending on how a forceful peer chooses the peers to connect with. In strategy D (resp. D^2, $1/D$), a node is chosen with a probability proportional to its degree (resp. the square of the degree, the inverse of the degree), while the choice is made uniformly in Strategy U. It is worth mentioning that when a normal node is chosen to be a neighbor, an edge is established with $weight = 1$. While if a neighbor is chosen again, the *weight* of the edge is incremented by 1. Moreover, to facilitate the interpretation of results, we assume that the weights of edges between normal peers are equal to 1 and the initial opinion of normal peers (the vector O_N) are set to 0. In other words, normal peers are initially neutral.

To examine the strategies mentioned above, we assign two different strategies to the opposing forceful peers (we will refer to forceful peers by their adopted strategies). Then we run a simulation of opinion dissemination as mentioned in Algorithm 1. The winning strategy is the one which has the majority of followers, with $neut\_thresh = 0.001$. For a fair judgment of the efficiency of strategies, we repeat the simulation 15,000 times for each strategy pair match. The value of α in Algorithm 1 is 0.3.

The matches are realized for all possible combinations of strategy pairs: U vs D, U vs D^2, U vs $1/D$, D vs D^2, D vs $1/D$, D^2 vs $1/D$. Matches between pairs are examined on different graph types maintaining a density ≈0.1 (excluding forceful peers). The number of Nodes is equal to 100. For illustration, we present results in a tabular form for matches between one pair of strategies (U against D) where Followers column is the average percentage of followers, and Wins column is the percentage of simulations won over the 15,000 simulations. To avoid confusion, the sum of wins of both opposing strategies can be less than 100% due to the matches where the followers of both strategies are equal. Then we present a figure that summarizes the results of the six matches combined by showing the average winning percentage each strategy had against all other strategies.

3.1 Geometric Graph

We first consider random geometric graphs, also known as disc graphs where nodes are uniformly distributed in a square. Each node is connected to all nodes within a circular vicinity centered at the node with a specified radius parameter. As mentioned above, the number of normal nodes is 100 and the density of the graph is around 0.1.

Table 1 details the outcome of strategies U vs D clash. The left part of Fig. 1 summarizes the results obtained for differents strategies by showing the order of dominance of strategies at each budget value. As shown in Fig. 1 for a small budget of 10, focusing on connecting to highly connected nodes proves to be the favored strategy. This can be seen as D and D^2 strategies have approximately the same effect. By Increasing the budget to 50, we can notice that the negative effect of over focusing on highly connected nodes is beginning to appear as D^2 loses its winning percentage to U. Not long after, D strategy follows D^2 at a budget of

Table 1. Geometric graph: U vs D

Budget	U followers	D followers	U wins	D wins
10	45.17%	47.61%	41.71%	55.37%
50	48.88%	49.50%	44.93%	50.31%
100	49.56%	49.48%	47.16%	46.83%
200	49.88%	49.43%	48.61%	44.18%
500	50.13%	49.30%	51.19%	40.80%

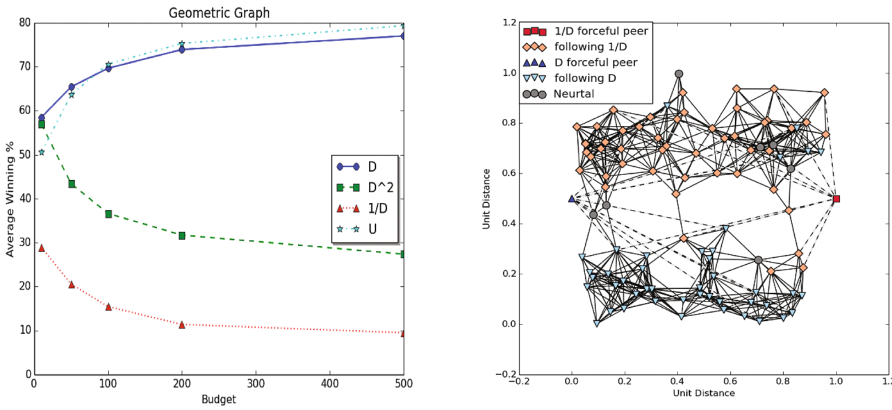

Fig. 1. Geometric graph: order of dominance and community structure

100 to leave U as the most effective strategy. Increasing the budget to 200 and 500, the negative effect of over focusing is mostly evident. An explanation of the effect of budget increase on strategies' efficiency can be related to the community structure exhibited by the geometric graph. The right part of Fig. 1 depicts a match between D and $1/D$ strategies, each with $budget = 10$. Noticeably, the graph can be divided into two major communities of nodes each condensed in a region and connected to the other communities by smaller number of nodes. Generally in geometric graphs, the number of communities is small and central nodes are highly connected, so D or D^2 strategies achieve good balance between spreading and connecting to highly connected nodes with small budget. As the budget increases, the focus on highly connected nodes becomes less and less efficient giving the advantage to strategy U.

3.2 Erdos-Renyi Graph

In contrast to the geometric graph, the Erdos-Renyi graph does not exhibit the community structure as it is more uniform. That is why it can be noticed that the negative effect of focusing on highly connected nodes appears in later stages. D^2 proves to be the most efficient with budget 10 exploiting all the available edges to connect to highly connected nodes. As shown by the left part of Fig. 2, at budget 50, D overcomes D^2 which implies that it achieves better balance between spreading in the network and connecting to high degree nodes. U strategy begins to exploit its spreading property at budgets of 100 and 200 until it overtakes the D strategy at a budget of 500.

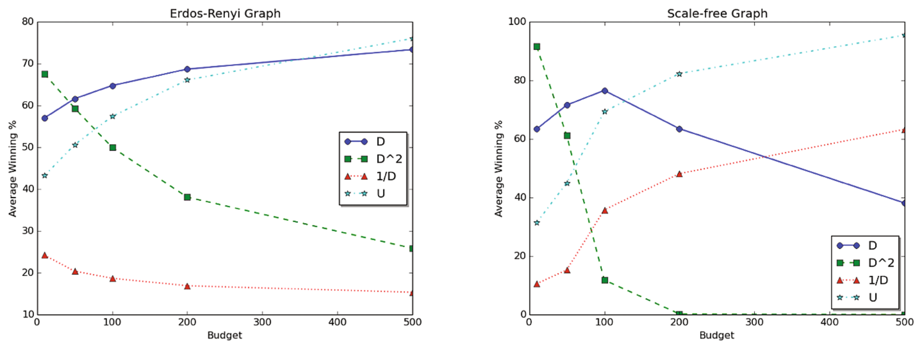

Fig. 2. Erdos-Renyi and sacle-free graphs: order of dominance of strategies

3.3 Scale-Free Graph

We have chosen barabasi-Albert as a model of scale-free graphs for our experiment. The Barabasi-Albert graph shows scale-free property with a power law degree distribution. Very few but very highly connected nodes are present due to the preferential attachment property that is acquired during the growth of the graph. The extremely high degrees for few nodes have a drastic effect on strategies and some results are quite unexpected (right part of Fig. 2).

Firstly, the $1/D$ which occupied the least position in all matches in previous graphs moves up the ranking with budget change. Moreover, unexpectedly the D^2 strategy which is the most effective for a budget of 10 turns to have a zero average winning percentage when the budget increases to 200. Similar to what happened in geometric graph but with a larger magnitude D^2 leverages on its few resources (budget) to target the most important nodes in the graph, followed by D, U then $1/D$. The increase of budget in this case shows more rapidly the negative effect of over-focusing on high degree nodes, letting the D^2 be the worst strategy for a budget of 100 surpassed by the $1/D$. The unanticipated improvement in $1/D$ can be explained also by the spreading of opinion factor. Since very few nodes are highly connected, the majority of nodes chosen by the $1/D$ strategy are efficiently balanced. At the budget of 200, D begins to suffer and U takes the first place. Finally, with a 500 budget D^2 is totally paralyzed and even D is surpassed by $1/D$.

3.4 Followers' and Winning Percentages

Winning percentage can be seen as the probability of winning against the opponent. Intuitively, the probability of winning increases as the difference in the followers' percentages increases. We only point out the main observation: even if the difference between two strategies in terms of number of followers is small, the difference between wining probabilities can be large. This is of course easy to understand since having just one more follower than his opponent is enough to win.

4 Knowledge Advantage

In this section we introduce the smart peer who has prior knowledge of connections established by the opposing peer. This knowledge is used by the smart peer to formulate a mixed integer linear program (MILP) in order to acquire the majority of followers with the least possible budget.

The strategy adopted by the other forceful peer is supposed to be known. Let λ^i be the number of connections from (non-smart) forceful peer to i^{th} node $\forall i \in N$. Let d^i be the total weight from i^{th} node to normal nodes only. Both d^i and λ^i are known. Let us now introduce the problem variables. Let x^i be the number of connections from smart peer to i^{th} node and let y^i be the final opinion of i^{th} node. A binary variable p^i is equal to 0 if the i^{th} node follows the smart peers, and 1 otherwise. We also use z^i_j binary variabes to represent x^i (see, (4b)). Some intermediate variables t^i_j are used to denote the products $(y^i z^i_j)$. The problem can then be formulated using the following MILP.

$$\min \sum_{i \in N} x^i \tag{4a}$$

$$x^i = \sum_{j=0}^{b} z_j^i 2^j \tag{4b}$$

$$y^i \left(\lambda^i + d^i\right) + \sum_{j=0}^{b} t_j^i 2^j = (1-\alpha) \left(x^i - \lambda^i + \sum_{k \in neigh(i)} y^k \right) \tag{4c}$$

$$-z_j^i \leq t_j^i \leq z_j^i \tag{4d}$$

$$y^i - 1 + z_j^i \leq t_j^i \leq 1 + y^i - z_j^i \tag{4e}$$

$$y^i \geq -1 + p^i (1 + neut\_thresh) \tag{4f}$$

$$\sum_{i \in N} p^i \geq \lceil \frac{N+1}{2} \rceil \tag{4g}$$

$$z_j^i, p^i \in \{0, 1\} \tag{4h}$$

The objective function (4a) aims to minimize the total budget used by the smart peer to win. The first constraint (4b) is a binary representation of the edge weights of the smart peer which will be used to conserve linearity of the problem. The b upper limit of the summation is the number of binary digits representing weight; it limits the maximum edge weight the smart peer can assign to a normal peer. The second constraint (4c) is derived from the opinion update equation mentioned in Algorithm 1 with the introduction of variables t_j^i to avoid variable multiplication $(y^i x^i)$. t_j^i represents the product $(y^i z_j^i)$ as ensured by (4d) and (4e). In constraint (4c), we used the fact that the initial opinion O_N is 0 and the weights of links between normal peers are equal to 1. p^i is defined in (4f), and used in (4g) to guarantee that more than half of the normal nodes follow the smart peer. Gurobi solver [21] is used. For all matches we set a time limit of one hour for the optimization problem computation after which the reached solution is returned by the solver.

Smart Peer Vs Other Strategies

From the four strategies mentioned above we run a simulation match between the smart peer against the most dominant strategy in each graph/budget configuration. Figure 3 shows the budget needed by the smart peer to win against the most dominant strategy in each graph configuration.

To quantify the budget needed by the smart peer to beat an existing forceful peer, we draw a best fit line. The best fit line shown as a solid line in Fig. 3 has the following equation: $f(x) = 0.5x + 4.85$, where x is the budget of the dominant strategy (non-smart) and $f(x)$ is the budget needed by the smart peer to win. The main conclusion is that with prior knowledge of opponent's connections, it is possible to win with nearly half of his/her budget.

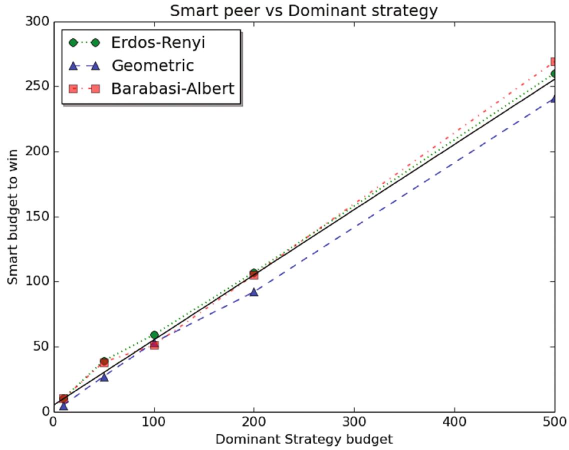

Fig. 3. Smart Peer against dominant strategies

5 Conclusion

We have shown that the best strategy to strongly influence the members of a social network depends on the underlying graph structure and the budget. Thus, for each budget/graph pair configuration, the proposed strategies can be ranked from worst to best. We also observed that knowing the opponent's strategy is a decisive advantage; the budget can be reduced by 50%!

This work can be extended in several ways. Strategies have been compared given the same budget. It might be interesting to study the robustness of the ranking of strategies by assuming that one strategy has slightly more budget than another. Other opinion propagation models can also be studied. One can for example consider the majority model where each node has a binary opinion following the opinion of the majority of his neighbors. Another worth-studying problem is the competition between more than two candidates. Finally, one can also consider antagonistic interactions inside the network represented by negative link weight.

References

1. Hendrikx, F., Bubendorfer, K., Chard, R.: Reputation systems: a survey and taxonomy. J. Parallel Distrib. Comput. **75**, 184–197 (2015)
2. Block Chain Principle. http://blockchain.info
3. Altman, E., Kumar, P., Venkatramanan, S., Kumar, A.: Competition over timeline in social networks. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1352–1357. IEEE (2013)
4. Neglia, G., Ye, X., Gabielkov, M., Legout, A.: How to network in online social networks. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 819–824. IEEE (2014)

5. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 539–550. Springer, Heidelberg (2010). doi:10.1007/978-3-642-17572-5_48
6. Chen, L., Leneutre, J.: Fight jamming with jamming-a game theoretic analysis of jamming attack in wireless networks and defense strategy. Comput. Netw. **55**(9), 2259–2270 (2011)
7. Stackoverflow. http://stackoverflow.com/
8. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: Proceedings of the 12th International Conference on World Wide Web, pp. 640–651. ACM (2003)
9. Bradai, A., Ben-Ameur, W., Afifi, H.: Byzantine resistant reputation-based trust management. In: 2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), pp. 269–278. IEEE (2013)
10. Twitter. http://twitter.com
11. Facebook. http://www.facebook.com
12. Chen, K., Liu, G., Shen, H., Qi, F.: Sociallink: utilizing social network and transaction links for effective trust management in P2P file sharing systems. In: 2015 IEEE International Conference on Peer-to-Peer Computing (P2P), pp. 1–10. IEEE (2015)
13. DeGroot, M.H.: Reaching a consensus. J. Am. Stat. Assoc. **69**(345), 118–121 (1974)
14. Krause, U.: A discrete nonlinear and non-autonomous model of consensus formation. In: Communications in Difference Equations, pp. 227–236 (2000)
15. Acemoglu, D., Ozdaglar, A.: Opinion dynamics and learning in social networks. Dyn. Games Appl. **1**(1), 3–49 (2011)
16. Ben-Ameur, W., Bianchi, P., Jakubowicz, J.: Robust distributed consensus using total variation. IEEE Trans. Autom. Control **61**(6), 1550–1564 (2016)
17. Ben-Ameur, W., Bianchi, P., Jakubowicz, J.: Robust average consensus using total variation gossip algorithm. In: 2012 6th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), pp. 99–106. IEEE (2012)
18. Erdos, P., Renyi, A.: On random graphs I. Publ. Math. Debrecen **6**, 290–297 (1959)
19. Penrose, M.: Random Geometric Graphs (No. 5). Oxford University Press (2003)
20. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
21. Gurobi optimizer reference manual. http://www.gurobi.com