

SCIENTIFIC REPORTS



OPEN

Influence Maximization for Fixed Heterogeneous Thresholds

P. D. Karampourniotis^{1,2}, B. K. Szymanski^{2,3,4}  & G. Korniss^{1,2}

Influence Maximization is a NP-hard problem of selecting the optimal set of influencers in a network. Here, we propose two new approaches to influence maximization based on two very different metrics. The first metric, termed Balanced Index (BI), is fast to compute and assigns top values to two kinds of nodes: those with high resistance to adoption, and those with large out-degree. This is done by linearly combining three properties of a node: its degree, susceptibility to new opinions, and the impact its activation will have on its neighborhood. Controlling the weights between those three terms has a huge impact on performance. The second metric, termed Group Performance Index (GPI), measures performance of each node as an initiator when it is a part of randomly selected initiator set. In each such selection, the score assigned to each teammate is inversely proportional to the number of initiators causing the desired spread. These two metrics are applicable to various cascade models; here we test them on the Linear Threshold Model with fixed and known thresholds. Furthermore, we study the impact of network degree assortativity and threshold distribution on the cascade size for metrics including ours. The results demonstrate our two metrics deliver strong performance for influence maximization.

Cascading processes emerge naturally through the interactions of nodes in different states in natural and human-made networks. Microscopic processes can potentially have large macroscopic impact on networks. In the case of human-made networks, their ever-increasing size and interconnectedness exponentially increases the uncanny impact of cascade processes. For instance, in financial or power grid networks, small initial perturbations or failures can result in cascades in the network causing tremendous disasters of global impact^{1–3}. In social networks, contact processes, namely social influence (or contagion), enable the spread of new behaviors, opinions and products, driving politics, social movements and norms, and viral marketing.

The key challenge in predicting such cascading processes is the identification of nodes whose change of state can potentially affect large portions of the networks. It is a computationally hard problem, and as such, multiple heuristics, theoretical analyses and algorithms have been introduced to solve it^{4–6}. Some are designed to address the specific nature of cascade processes, while others are based on more general algorithmic approaches or network based centrality measures. Such algorithms can be used to minimize disasters by, for example, re-enforcing weak nodes in power-grid nodes^{7–9}, or placing sensors to detect the contamination in water pipe network¹⁰. Likewise, arresting the spread of infectious disease requires a global awareness of its existence^{11–14}. Understanding cascades is also important for optimizing viral marketing^{15–17}. Yet, it is challenging to find the set of initiators (also called seeds) which, when put into a new state (opinion/idea/product), will maximize the spread of this state^{18–28}.

Here, we study the problem of influence maximization for a classical opinion contagion model, namely the Linear Threshold Model (LTM)^{29,30}. Yet our methods can be used for any percolation-based model. The LTM is designed to capture the peer pressure dynamics that lead an individual to accept a new state being propagated. It is a binary state model, where a node i has either adopted a new product/state/opinion, $n_i = 1$, or not, $n_i = 0$. In the LTM, each node in the network has a fractional threshold, an intrinsic parameter representing the node's resistance to peer pressure. The spreading rule is that an inactive node ($n_i = 0$), with in-degree k_i^{in} and threshold ϕ_i , adopts a new opinion only when the fraction of its neighbors $j \in \partial i$ holding the new opinion is higher than the node's threshold, that is $\sum_{j \in \partial i} n_j \geq \phi_i k_i^{\text{in}}$. The process is deterministic and once a node adopts a new opinion it cannot return to its previous state. The integer number of active neighbors required for node i to get active is given

¹Department of Physics, Applied Physics, and Astronomy, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, 12180-3590, USA. ²Social Cognitive Networks Academic Research Center, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, 12180-3590, USA. ³Department of Computer Science, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, 12180-3590, USA. ⁴Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland. Correspondence and requests for materials should be addressed to P.D.K. (email: karamp.pan@gmail.com)

by its initial resistance $r_i = \phi_i k_i^{\text{in}}$. Naturally, the maximum resistance of a node is k_i^{in} . At each step of the spread, the resistance is decreased by the number of the node's neighbors that adopted a new opinion. A node gets activated through spread when its resistance drops to zero or below, $r_i \leq 0$. Bootstrap percolation^{31,32} is an alternative formulation of the LTM where the thresholds are not fractional, but integer (resistance). Interestingly, LTM and bootstrap percolation conceptually share similarities with the integrate-and-fire-neuron model^{33–36}, with the main differences being that in the later, activation of a node is probabilistic rather than a fixed threshold value, and return to the initial (inactive) state $n_i = 0$ is allowed. The size of cascades in the LTM is governed by the thresholds of the nodes^{37,38}, the size of the initiator set³⁹, the selection of initiators²⁸, and of course, the structure of the underlying network^{40–44}.

Examples of LTM-type spread mechanisms and of the heterogeneity of the thresholds are provided through a number of controlled experiments^{45–47} and by empirical data analysis^{48–52}. Unicomb *et al.*⁵³ studied the threshold model in weighted networks and found that the time of cascade emergence depends non-monotonically on weight heterogeneities. Watts and Dodds³⁸ showed through simulations of various types of spread mechanisms that the cascade size is governed not by superspreaders, but by a small critical set of nodes with low resistance to influence. Karampourniotis *et al.* showed that the threshold distribution is important for the overall dynamics³⁷. In particular, with an increasing standard variation σ of thresholds (while holding the average of the thresholds fixed) a smaller initiator fraction is required for global cascades. Furthermore, they showed that in the vicinity of $\sigma \approx 0$ a tipping point appears as the required fraction of randomly selected initiators gradually increases. Yet, with gradually increasing variance σ , eventually the tipping point is replaced by smooth transition. In addition, Watts and Dodds³⁸ showed that a critical fraction of nodes with high susceptibility contributes to social influence much more than initiators with high network centrality.

Hence, the knowledge of thresholds or their distribution is critical for influence maximization algorithms. In the case of zero information about the thresholds or their distribution, a good assumption to make is that the threshold distribution is uniform. This is a very interesting case for which the spread function is submodular, that is, it follows a diminishing returns property¹⁸, which can be used for maximizing the influence^{10,18–21,24,54,55}. In general, submodularity does not hold when the thresholds are known and fixed or for any threshold distribution but the uniform. For the cases of fixed known thresholds, the influence of any seed set can be computed in polynomial time⁵⁶. Further, in the special case of a unique threshold for all nodes, a first-order transition appears^{39,57,58}. Then a powerful algorithm, namely CI-TM with complexity $\mathcal{O}(N \log N)$, provides great performance²⁸. In ref.⁵⁹ the authors express the influence maximization problem as a constraint-satisfaction problem and use believe propagation to solve it. Yet, for the case of identical thresholds it does not perform as well as the CI-TM with which it was compared in²⁸. Other analytical based metrics show the importance of network structure, but only for a small number of initiators⁶⁰. Furthermore, in⁶¹ the authors propose to use of an evolutionary algorithm implemented with general-purpose computing on graphics processing units (GPGPU) to tackle the challenge of combinatorics, at the additional cost of higher time and memory complexity. The authors show that their approach clearly outperforms the greedy algorithm for known thresholds both in cascade size and time, but it is currently limited to graphs of size in the order of $N = 10^3$.

Here, we study the efficiency of known selection strategies for fixed (and known) heterogeneous thresholds generated from different threshold distributions, and a range of assortativity values. We introduce two new and very different selection strategies for the LTM with fixed thresholds and compare them in terms of their performance with a number of other strategies, including the CI-TM, and greedy. Since we focus on fixed (and known) thresholds we do not include the performance of various network centrality measures like the Page Rank and k -core³⁹, which do not take into account the provided threshold information and thus are outperformed by the strategies that do.

Selection Strategies

We use many fast heuristics, which take advantage of the knowledge of thresholds. Since the thresholds are fixed and known, the cascade size caused by an initiator is deterministic. Hence, we sequentially introduce initiators *on the inactive subgraph* of the original network. First, the node with the highest dynamic fractional threshold (*thres*) is a reasonable choice. Likewise, a natural selection is the node with the highest dynamic out-degree k_i^{out} (i.e., counting only edges to currently inactive nodes) at each step (*deg*). Another possible heuristic for the LTM is the selection of the node with the highest resistance (*res*) at each step. Resistance r_i is the current (dynamic) integer threshold of node i , defined as the number of active neighbors required for the node to get activated. Accordingly, when a node is activated by a cascade or by being selected as an initiator, its resistance turns zero, and so a fully activated network has total resistance of zero.

The selection of any inactive node i as an initiator lowers the resistance of all its inactive neighbor nodes by one, for a total decrease of k_i^{out} . Moreover, if any neighbor of i had resistance equal to one, it will also get activated, further reducing the resistance of other nodes. In addition, if node i is the initiator, then its resistance is reduced to zero, $r_i = 0$. Therefore, the total reduction of the entire graph resistance is at least $r_i + k_i^{\text{out}}$. To capture the resistance drop at the direct neighbors of node i , termed direct drop (DD), we introduce the heuristic strategy DD, with metric $\text{DD}_i = r_i + k_i^{\text{out}}$. In addition, we introduce the heuristic method indirect drop of resistance (ID), by summing the DD metric and the drop of the resistance of the inactive subgraph caused by the neighbors of the node i which are indirectly activated by the selection of i as seed. That is $\text{ID}_i = r_i + k_i^{\text{out}} + \sum_{j \in \partial i | r_j = 1} (k_j^{\text{out}} - 1)$. The total drop of the network's resistance caused by choosing i as an initiator is at least as high as ID_i ; it is more if the spread expands beyond the direct neighborhood (e.g the second or further neighborhoods). The ID heuristic is equivalent to the CI-TM algorithm²⁸ for a sphere of influence $L = 1$ with the addition of the resistance, r_i , of the selected node i in the metric. For very large L , a node's i CI-TM score is essentially (assuming a tree-like approximation of the network) equal to the drop of the network's total resistance if i was the seed, minus node's i

resistance r_i . The metric of CI-TM is governed by the out-degree of the nodes surrounding the target node ignoring the challenge of activating nodes with high resistance and/or low in-degree. For comparison to our methods, we apply the CI-TM algorithm itself (for $L = 6$), and the greedy algorithm for fixed thresholds, where at each step the node which causing the maximum cascade size is selected.

Balanced Index Strategy. Constructing a selection strategy mainly based on the network structure or just on the resistance of the nodes, is not ideal, since useful information is not being utilized. On one hand, selecting nodes solely based on some network centrality metric, leads to many easily susceptible nodes being selected as initiators, nodes that could potentially be activated through spread. On the other hand, aiming at selecting high resistance nodes, does remove spread obstacles but does not guarantee they will be great influencers. The DD and ID strategies aim to address this weakness by using intuitive heuristics. To quantify the interplay between the importance of low resistance and high centrality nodes, we introduce the Balanced Index (BI) selection strategy. For this strategy, we assign weights (a, b, c) to each term of ID_i to capture the importance of each feature, that is

$$BI_i = ar_i + bk_i^{\text{out}} + c \sum_{j \in \partial(i) | r_j=1} (k_j^{\text{out}} - 1), \quad (1)$$

where $a + b + c = 1$ and $a, b, c \geq 0$. The optimal weights for influence maximization are determined by scanning within the ranges of weights in the ensemble of graphs and for various threshold distributions. In this case, the out-degree (deg) strategy corresponds to $(a, b, c) = (0, 1, 0)$, res to $(a, b, c) = (1, 0, 0)$, the CI-TM for $L = 1$ to $(a, b, c) = (0, 1/2, 1/2)$, while the two heuristics we introduced, DD_i and ID_i , correspond to weights $(a, b, c) = (1/2, 1/2, 0)$ and $(a, b, c) = (1/3, 1/3, 1/3)$ respectively. Interestingly, the weighted metric BI_i can be viewed as a measure (units) of resistance, however, in general, it does not correspond to the network's total resistance drop when i is the seed. As far as the time complexity of each method is concerned $\mathcal{O}(\langle k \rangle N + N \log N)$ the computation of a seed's (or set of seeds) induced spread takes $\mathcal{O}(\langle k \rangle N)$ time. Yet, (similar to²⁸) when computing the spreading process for sparse graphs, we can place a stop condition on the algorithm “ L levels” away from the seed node, reducing the complexity by $\mathcal{O}(N)$. In addition, using a heap structure, re-ordering the highest BI nodes takes $\mathcal{O}(N \log N)$. Thus, the time complexity of these methods for sparse graphs reduces to $\mathcal{O}(N \log N)$.

Group Performance Index algorithm. All of the above strategies are essentially local in nature, since they aim to maximize the number of activated nodes or to reduce the total resistance of the system caused by one initiator at a time. They lack in their metrics the impact of the combination of initiators on influence maximization, which by default limits their performance. Algorithms that use combinations of nodes in their metrics can lead to high-quality approximate global solutions. However, look-ahead algorithms suffer from the potentially prohibitive computational costs. For instance, to measure the total impact of a subset of g nodes selected from the total of N nodes, a brute force algorithm would require $\binom{N}{g}$ selections of possible initiator sets. Its complexity would be $\mathcal{O}(\langle k \rangle N^{g+1})$, which would be prohibitively computationally expensive except for very small g .

Instead, a probabilistic look-ahead algorithm would aim to reduce the number of combinations by randomly selecting initiators. That is, at each step t , in order to measure the impact of a node i in the presence of other nodes, i would have to be selected as an initiator. Then, the remaining initiators would be randomly selected. This process would be repeated ν times, each time recording the cascade size. We would have essentially measured the impact of node i as an initiator in the presence of randomly selected sets of initiators. We would repeat this process for all other inactive nodes, and finally select the node with the highest impact.

Here, we introduce (Alg. 1) the Group Performance Index algorithm (GPI). With GPI we target the nodes which, when included in any randomly selected initiator set (group), cause the group to have higher than average performance. GPI shares similarities with the probabilistic look-ahead, however it is much more efficient. First, we take advantage of the property that permutations of any set of initiators do not impact the total cascade size returned by that set for the LTM. By not having to scan each node individually when computing its impact in the presence of other initiators, we reduce the number of computations by $\mathcal{O}(N)$. Moreover, for the probabilistic look-ahead algorithm we would be selecting each initiator one-by-one, each time having to update the impact of each node by re-running simulations.

We select $\lceil sN \rceil$ initiators (instead of one-by-one, where $s \in (0, 1)$) defines granularity of initiator set increases, thus reducing the complexity by another $\mathcal{O}(N)$ factor. Also, when randomly picking nodes as test-initiators for our metric, we do not predefine the order of selecting initiators, but we pick them randomly one by one from the inactive nodes. Finally, we aim to maximize the cascade size for a specific number of initiators, which is essentially our cost budget. However, the impact of even a small fraction of initiators can potentially have a large impact on the cascade size, especially near a tipping point. That means, a nearby tipping point can be missed because of fixed seed budget. To address this, we aim to minimize the size of the initiator set in order for the cascade size to be at least as large as a specific predefined fraction of nodes, S_{goal} . However, in general, GPI can also be used when constraining either the seed budget, or computational time.

Let us start with the initial graph $G(V, E, r)$, where $V(G)$ is the set of N nodes, $E(G)$ is the set of edges of the graph, while r_i is the resistance of each node $i \in V(G)$. Our goal is to find the initiator set Y such that the fraction of nodes activated by it satisfies the inequality $f(Y)/N \geq S_{\text{goal}}$, where the function $f(X)$ expresses the number of nodes that got activated from the set X (here, we also include the initiator nodes in $f(X)$), see Alg. 1. To do so, at each step t we select the set Q of $\lceil sN \rceil$ nodes with the highest GPI-ranking (we will define it below) as initiators. Then, we compute the cascade induced by Q on the graph with cascade induced by nodes in Y already included in graph G and we include the Q set of nodes in the initiator set Y by updating $Y = Y \cup Q$. Finally, we update the spread size, and update $G(V, E, r)$ (reducing the resistance, and removing all activated nodes and their edges).

As stated above, at any step t we look for the nodes with the highest GPI value, that is, the nodes which, when present in the initiator set, cause the desired cascade size to be reached (on average) with a small initiator set size. To measure the expected GPI, a large enough number ν of simulations is required. Each simulation j is run on a series of graphs with the initial one defined as $G_{\text{test}} = G$. The simulation stops when the desired cascade size $\lceil S_{\text{goal}}N \rceil$ is reached. For each simulation j , we randomly select one-by-one an inactive node, place it on the initially empty set $Y_{\text{test},j}$ then we compute the spread, and update G_{test} accordingly. Each simulation stops when $f(Y_{\text{test},j}) \geq S_{\text{goal}}N - f(Y)$.

Let $n_{j,i}$ be 1 when $i \in Y_{\text{test},j}$, and 0 otherwise, informing whether node i belongs to set $Y_{\text{test},j}$. At the beginning of each simulation, $Y_{\text{test},j}$ is empty, hence, $n_{j,i}$ for each node i is initially zero. Then, for each node i that belongs to $Y_{\text{test},j}$, GPI metric can be expressed analytically as

$$\text{GPI}_i = \frac{\sum_{j=1}^{\nu} |Y_{\text{test},j}| n_{j,i}}{\sum_{j=1}^{\nu} n_{j,i}}, \text{ for } i \text{ such that } \sum_{j=1}^{\nu} n_{j,i} > 0 \tag{2}$$

The numerator of GPI, is the sum of sizes of the randomly selected initiator sets $|Y_{\text{test},j}|$ in which node i is present. Since we select inactive nodes uniformly at random, the nodes do not appear in the initiator sets equal number of times. If that was not the case (that is if all nodes were equally frequently chosen just like for the probabilistic look-ahead we mentioned above), the numerator of the fraction of Eq. 2 would be a sufficient metric, where the smaller the sum is the larger the impact of node i is. The presence of the denominator is necessary to normalize the number of times node i is selected as an initiator. In addition, because we only select inactive nodes, the nodes which are more likely to be activated through spread, that is typically nodes with low resistance and high in-degree, are going to be less frequently a part of the initiator set than others. And so, nodes with a large number of appearances are nodes less likely to be activated by diffusion than others.

Since GPI deals with the expected impact of nodes, it is by default slower than the rest of the strategies but can potentially find much better initiator sets than other strategies do. Computing the GPI of each node takes $\mathcal{O}(\nu \langle k \rangle N)$, since, we randomly select one-by-one test-initiators and compute the spread $\mathcal{O}(\langle k \rangle N)$ (which for sparse graphs becomes $\mathcal{O}(N)$) until the S_{goal} is satisfied, and finally we repeat the process ν times. Sorting the nodes based on their GPI ranking takes $\mathcal{O}(N \log N)$ using a heap structure. Finally, for each Q , computing the spread induced from the highest GPI-ranked takes $\mathcal{O}(\langle k \rangle N)$. Thus, the complexity for each batch of seeds Q is $\mathcal{O}(\nu \langle k \rangle N + N \log N + \langle k \rangle N)$, so $\mathcal{O}(\nu \langle k \rangle N + N \log N)$ for $\nu \gg 1$. Since the final size of the seed set $|Y|$ is $\mathcal{O}(N)$, we require $\lceil |Y| / \lceil sN \rceil \rceil$ steps which is $\mathcal{O}(1/s)$, hence the final complexity is $1/s * \mathcal{O}(\nu \langle k \rangle N + N \log N)$. From our experience, for a good estimation of the expected GPI value of each node the number of randomizations ν should be of the order of the system size N , $\mathcal{O}(N)$. Thus, the worst case time complexity becomes $1/s * \mathcal{O}(\langle k \rangle N^2 + N \log N)$. In addition, it's possible to control the size of the computation of the spread any initiator would induce on a sphere

Algorithm 1. Group Performance Index Algorithm.

procedure GROUP PERFORMANCE INDEX ALGORITHM

Input Graph $G(V, E, r)$ and thresholds ϕ_i for each node i
 Input the desired cascade fraction S_{goal} , step granularity fraction s , number of randomizations ν
 Initialize the initiator set $Y = \emptyset$ and cascade fraction $S_y = 0$
 Get the resistance of each node, $r_i = \lceil \phi_i k_i^{\text{in}} \rceil$
 Initiate step counter $t = 0$
while $S_y < S_{\text{goal}}$ **do**
 $t \leftarrow t + 1$
 Initialize GPI_i ▷ average impact GPI for node i
 Initialize nu_i ▷ numerator of Eq. 2 for node i
 Initialize de_i ▷ denominator of Eq. 2 for node i
 Initialize $Q = \emptyset$
 for $j = 1 : \nu$ **do**
 $G_{\text{test}} = G$
 Initialize $Y_{\text{test}} = \emptyset$ ▷ set of test-initiators
 Initiate the local test cascade fraction $S_l = 0$
 while $S_l < S_{\text{goal}} - S_y$ **do**
 Randomly select an inactive node i as test initiator
 $Y_{\text{test}} \leftarrow Y_{\text{test}} \cup \{i\}$ ▷ Add node i to the initiator set Y_{test}
 Run on G_{test} the cascade induced by i :
 Compute the additional cascade size $f(i)$, and update S_l , $S_l \leftarrow S_l + f(i)/N$
 Add all newly activated nodes to Y_{test}
 $de_{i \in Y_{\text{test}}} \leftarrow de_{i \in Y_{\text{test}}} + 1$, $nu_{i \in Y_{\text{test}}} \leftarrow nu_{i \in Y_{\text{test}}} + |Y_{\text{test}}|$
 Update graph G_{test} (update node resistance, remove the activated nodes and their edges)
 $\text{GPI}_i = nu_i / de_i$, for all $i \in Y_{\text{test}}$
 Insert the top $\lceil sN \rceil$ GPI-ranking nodes to Q , and Y , updating $Y \leftarrow Y \cup Q$
 Run the cascade on G induced by the Q :
 Compute the additional cascade size $f(Q)$ and update $S_y \leftarrow S_y + f(Q)/N$
 Update graph G (update node resistance, remove activated nodes and their edges)

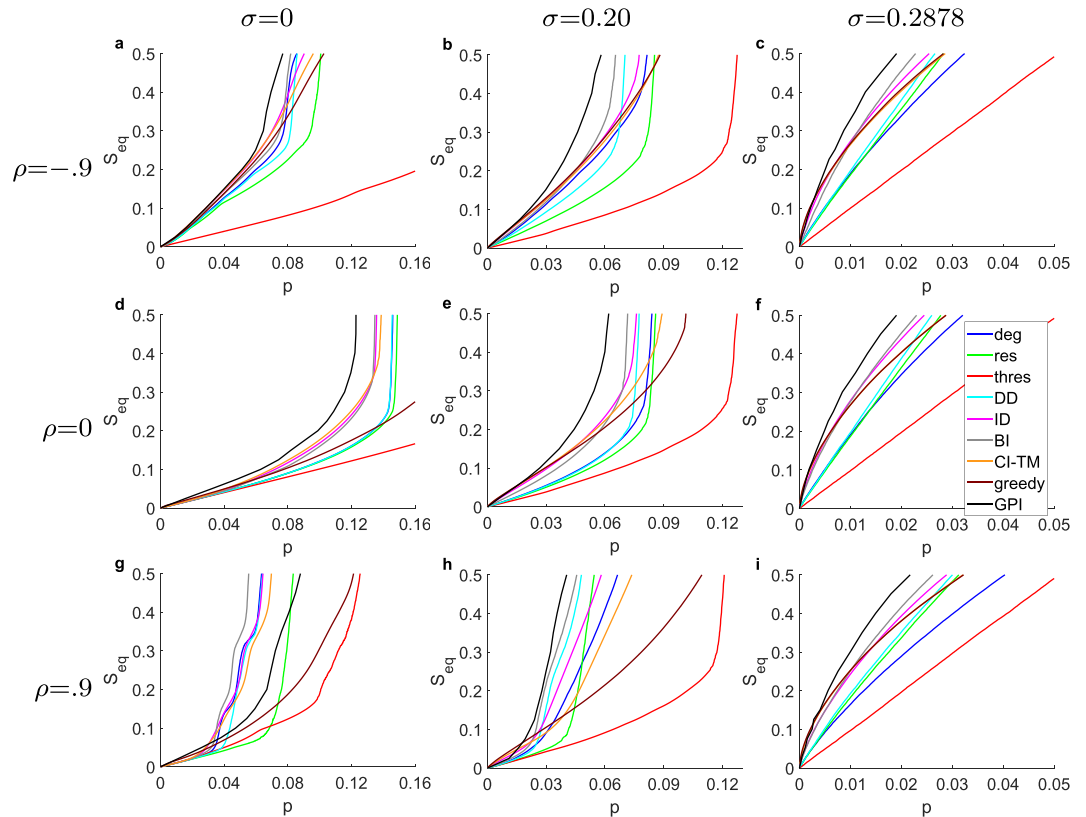


Figure 1. Comparison of the BI and GPI (for different S_{goal} selection strategies in terms of cascade performance S_{eq} for (a–c) $\rho = -0.9$, for (d–f) $\rho = 0$, for (g–i) $\rho = 0.9$, for (a–g) $\sigma = 0$, for (b–h) $\sigma = 0.20$, for (c–i) $\sigma = 0.2887$, with $\bar{\phi} = 0.5$, averaged for 500 different network realizations (except for the GPI which is based on 20 realizations) each with a different threshold generation, applied on ER graphs with $N = 10,000$ and $\langle k \rangle = 10$. For BI, the weights are optimized for $S_{eq} = 0.5$. For GPI, the fraction of top s initiators, and number of randomizations are respectively $s = 0.001$, and $\nu = 100,000$.

of influence L (see²⁸). Then, under the assumption of sparse graphs the spread computation is an $\mathcal{O}(1)$ process, hence the total complexity becomes $1/s * \mathcal{O}((k)N + N \log N)$; which, for sparse graphs with average degree independent of the system size N , it becomes $1/s * \mathcal{O}(N \log N)$. Hence, GPI becomes competitive in terms of complexity with the best-known solutions. Furthermore, for the case of an identical threshold for all nodes, since there is a sharp phase transition point (and small spread otherwise), the computation of spread size is even faster.

For the remainder of the paper, unless otherwise specified, we set the control parameters of GPI as $s = 10^{-3}$, $\nu = 10^5$, and $S_{goal} = 0.5$, and will compute the full spread size induced by any initiator node. Those value were chosen based on performance of alternatives. Indeed, ν/s defines the computational budget for finding a solution, so for fair comparison of different values of ν and s , we need to keep ν/s constant. Hence, a smaller value of s means that ν needs to be kept low resulting in poor selection of seeds due to unreliable estimate of GPI. In contrast, large s means that we have large ν which results in good selection of seeds, but also large Q , so these seeds are likely to become obsolete before they are replaced because graph may change much after less than $|Q|$ seeds are added. Finally, when $s > 1/N$, then there is a non-zero probability that a (small fraction) of selected initiators will be activated from a previously selected initiator. For those cases, we excluded the activated initiators from the final initiator set.

Results

We are comparing the performance (cascade size S_{eq}) of the strategies for the entire parameter space of network assortativity ρ , and threshold distribution with fixed average threshold $\bar{\phi} = 0.5$ and varying standard deviation σ . In Fig. 1 we present our main results for the ensemble of ER graphs for the extreme cases of high positive ($\rho = 0.9$) and high negative ($\rho = -0.9$), and neutral ($\rho = 0$) assortativity, measured with Spearman's rank correlation coefficient ρ^{62} . Furthermore, we examine the cases of a identical thresholds ($\sigma = 0$), a uniform threshold distribution ($\sigma = 0.287$), and a truncated normal distribution in between ($\sigma = 0.2$). For the GPI strategy we present the critical initiator fraction for which the S_{goal} is satisfied. First, focusing especially on ER graphs ($\rho = 0$) we notice that as we move from a threshold distribution with standard deviation $\sigma = 0$ to larger σ , there is a change from a first-order phase transition to a smooth crossover also seen for randomly selected initiators in³⁷. Interestingly, in the case of the uniform threshold distribution ($\sigma = 0.287$) at the ensemble level, we observe that all the direct methods appear to have diminishing returns with increasing cascade size, that is, the contribution to the cascade size of any additional initiator in an initiator set is diminishing as the initiator set is becoming larger.

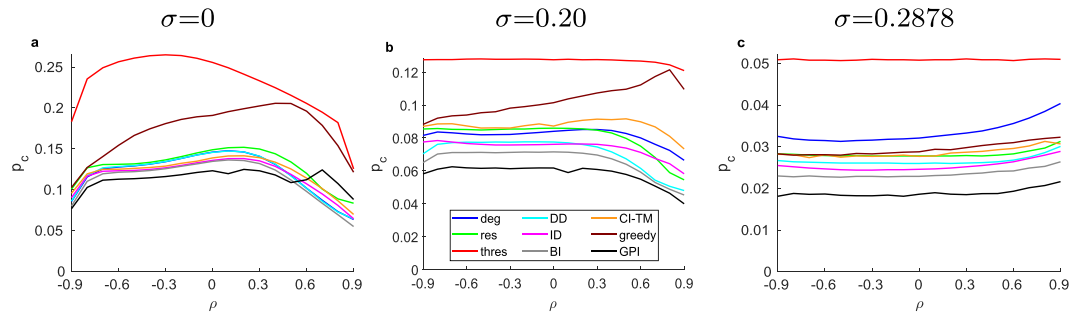


Figure 2. Initiator fraction p_c required to reach spread $S_{eq} = 0.5$ vs. degree assortativity ρ for graphs with $N = 10,000$ and $\langle k \rangle = 10$ for (a) $\sigma = 0$, for (b) $\sigma = 0.20$, and for (c) $\sigma = 0.2887$ with $\bar{\phi} = 0.5$, averaged for 500 different network realizations (except for the GPI which is 20) each with a different threshold generation.

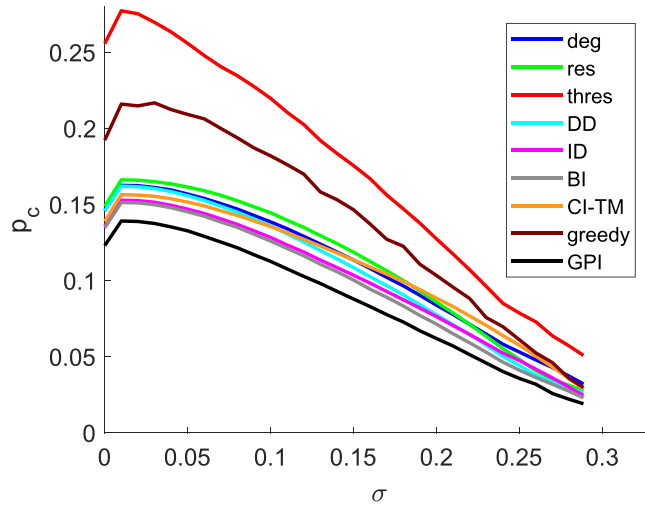


Figure 3. Initiator fraction p_c required to reach spread $S_{eq} = 0.5$ vs. the standard deviation σ of the generated threshold distribution with $\bar{\phi} = 0.5$ for graphs with $N = 10,000$, $\langle k \rangle = 10$ and a $\rho = -0.9$, b $\rho = 0$, c, $\rho = 0.9$, averaged for 500 different network realizations (except for the GPI which based on 20 realizations) each with a different threshold generation.

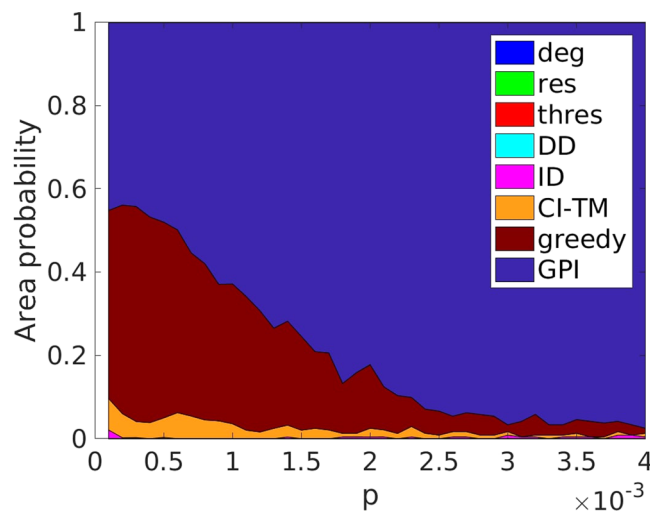


Figure 4. Probability of each strategy being the best strategy for one network with $N = 10,000$, $\langle k \rangle = 10$, $\rho = 0$, $\sigma = 0.2887$, and $\bar{\phi} = 0.5$, for 240 threshold generations (same for each strategy).

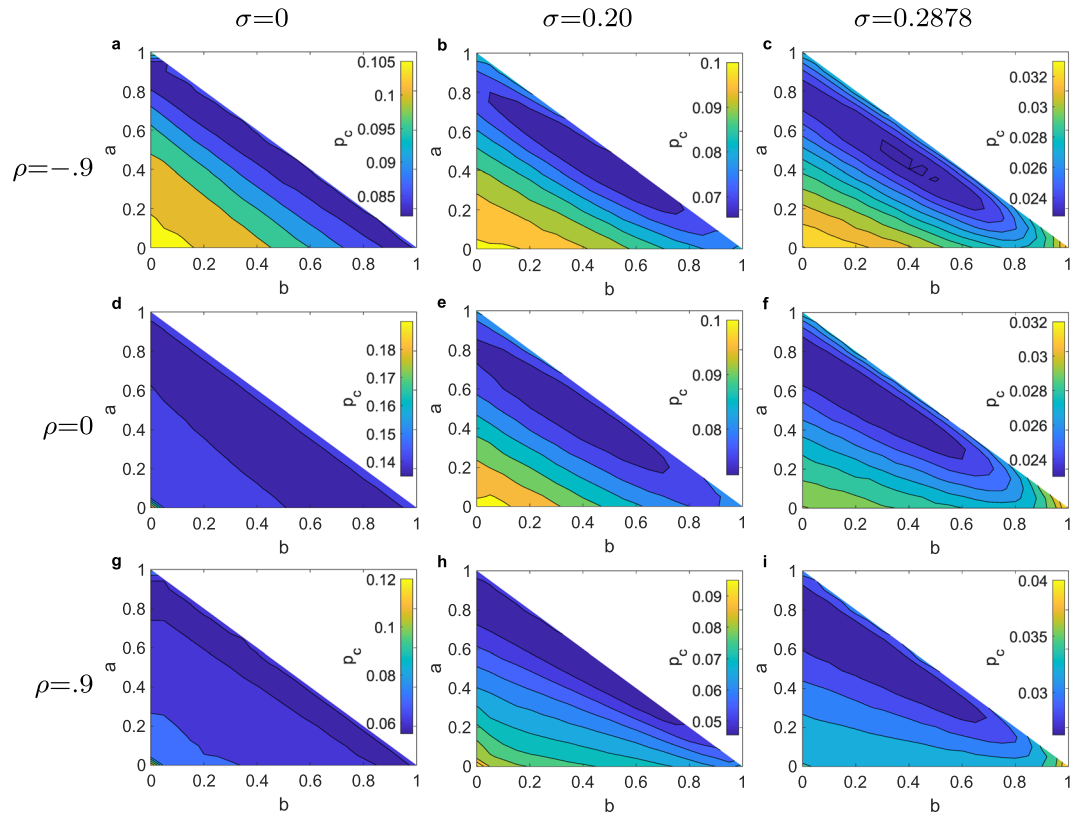


Figure 5. Contours of p_c obtained by the BI selection strategy for reaching $S_{\text{eq}} = 0.5$ by controlling parameters a and b (from Eq. (1), with $a + b + c = 1$) for graphs with $N = 10,000$, $\langle k \rangle = 10$ and for (a–c) $\rho = -0.9$, for (d–f) $\rho = 0$, for (g–i) $\rho = 0.9$, for (a–g) $\sigma = 0$, for (b–h) $\sigma = 0.20$, for (c–i) $\sigma = 0.2887$, with mean threshold $\bar{\phi} = 0.5$, for 500 repetitions, averaged for 500 different network realizations each with a different threshold generation. The resolution in the a and b weight space are 0.05.

As far as the performance of the strategies is concerned, the degree (*deg*) strategy's relative performance is decreasing for larger σ 's, while the resistance strategy's performance is increasing. In addition, CI-TM which incorporates a network structure decomposition using the neighboring nodes with the information about resistance $\nu = 1$ in it as metric, is out-performing the degree strategy for the case of $\sigma = 0$ and $\rho = 0$ but it is not performing as well in the rest of cases. On the other hand, the ID approach is outperforming the degree, the resistance and the CI-TM strategy for all cases of ρ and σ . Naturally, the introduced weighted strategy is outperforming in all cases the strategies that incorporate their ranking metrics (*deg*, *res*, CI-TM, DD, ID), especially for $S_{\text{eq}} = 0.5$ which is what we optimized it for here.

The GPI strategy largely outperforms the rest of the strategies in all cases except for the case of very high assortativity ρ with identical thresholds ($\sigma = 0$). Yet, for a desired cascade size of $S_{\text{eq}} = 0.5$ with even lower s and higher ν the performance of GPI improves. The strong performance of GPI in the case of ER graphs ($\rho = 0$) with an identical threshold $\sigma = 0$ reveals the importance of the combination of initiators for Influence Maximization. We have used a $s = 10^{-3}$ fraction of initiator and $\nu = 10^5$, even though the best resolution we could achieve for a graph with size N would be $s = 1/N$, because then the initiators are inserted one-by-one. Additionally, we demonstrate the performance of our strategies against ER graphs with varying average degree, as well empirical networks (see Figs 1 and 2 of the Supplementary Information).

To further study the direct methods, we evaluate their performance in the assortativity space ρ (Fig. 2) and the threshold distribution space σ (Fig. 3) for $S_{\text{eq}} = 0.5$. Evaluating through ρ for $\sigma = 0$, we observe a highly non-monotonic behavior for all the strategies. Between the range of approximately $-0.8 \leq \rho \leq 0.4$, CI-TM is outperforming the degree strategy, which is approximately in the same regime, in which ID is outperforming all other direct strategies. Yet, for $\rho \geq 0.5$ DD is the best strategy. With increased deviation of the thresholds $\sigma = 0.20$ (Fig. 2b), the performance of the strategies which depend more on the network structure like the degree and CI-TM is getting worse, while strategies which give higher importance to the resistance of the node, like *res*, DD and ID are performing better. Finally, for a uniform threshold distribution $\sigma = 0.2887$ (Fig. 2c), we observe that strategies show a convex response to ρ , with ID being the best strategy (for a desired cascade size $S_{\text{goal}} = 0.5$). On the other hand, the threshold (*thres*) strategy appears to be independent of ρ for thresholds generated randomly with $\sigma = 0.2887$. Finally, from Fig. 3 we observe that ID has the best performance compared to all the direct strategies over nearly all of the entire range of σ , making it the best overall strategy (excluding the BI and GPI strategies).

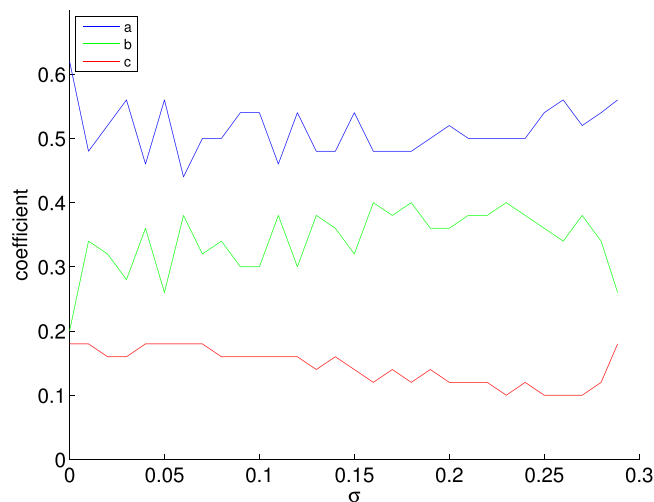


Figure 6. Impact of standard deviation σ on the optimal weights of BI (from Eq. (1), with $a + b + c = 1$) for desired cascade $S_{\text{goal}} = 0.5$, for ER graphs with $N = 10,000$, $\langle k \rangle = 10$, $\rho = 0$, with $\phi = 0.5$, averaged for 500 different network realizations each with a different threshold generation. The resolution in the a and b weight space are 0.02.

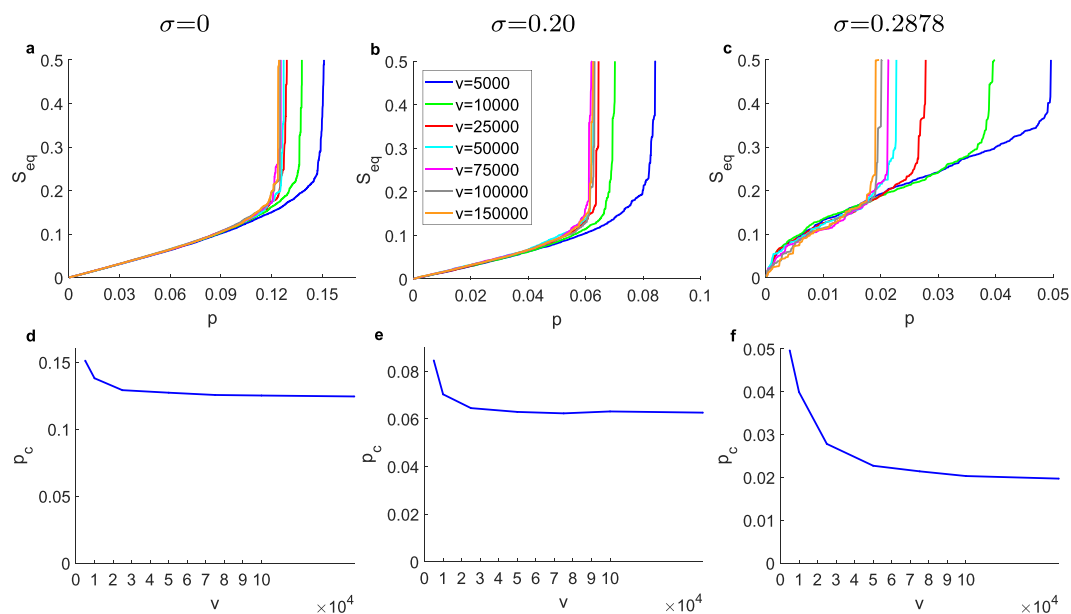


Figure 7. Impact of the number of randomizations ν on the performance of the GPI strategy for $s = 0.0025$ for desired cascade $S_{\text{goal}} = 0.5$, for ER graphs with $N = 10,000$, $\langle k \rangle = 10$, $\rho = 0$. (a–c) Cascade S_{eq} vs. the initiator fraction p for $\sigma = 0, 0.2, 0.2887$, respectively (for one realization). (d–f) Initiator fraction p_c required for desired cascade $S_{\text{goal}} = 0.5$ vs. randomizations ν for $\sigma = 0, 0.2, 0.2887$, respectively (for one realization).

In order to better study the performance of each strategy, we additionally examined the probability of any strategy being the best one (Fig. 4) for the same initiator fraction p . In particular, we tested the performance of all strategies at a fixed randomly generated ER network, for 200 different threshold generations. The actual cascade size S_{eq} vs. initiator fraction p can be seen in the Supplementary Information Fig. 3a (average) and Fig. 3b (50 runs). GPI remains superior even for a very small number of initiators.

First, we notice that the greedy algorithm is in all cases outperforming all other strategies for a small initiator fraction, while for a very large initiator fraction all strategies have the same performance, because all of the network is activated. For an ER graph ($\rho = 0$) with $\sigma = 0$ the spread is minimal until the phase transition point has been reached, and CI-TM leads until the tipping point for ID has been reached. Since the metric of CI-TM takes into account the out-degree of the nodes, and does not consider the effort required to activate nodes with high resistance and low in-degree, it is outperforming other methods for smaller initiator sizes, but eventually gets surpassed, when the nodes with high resistance with a structural importance have not been activated. However,

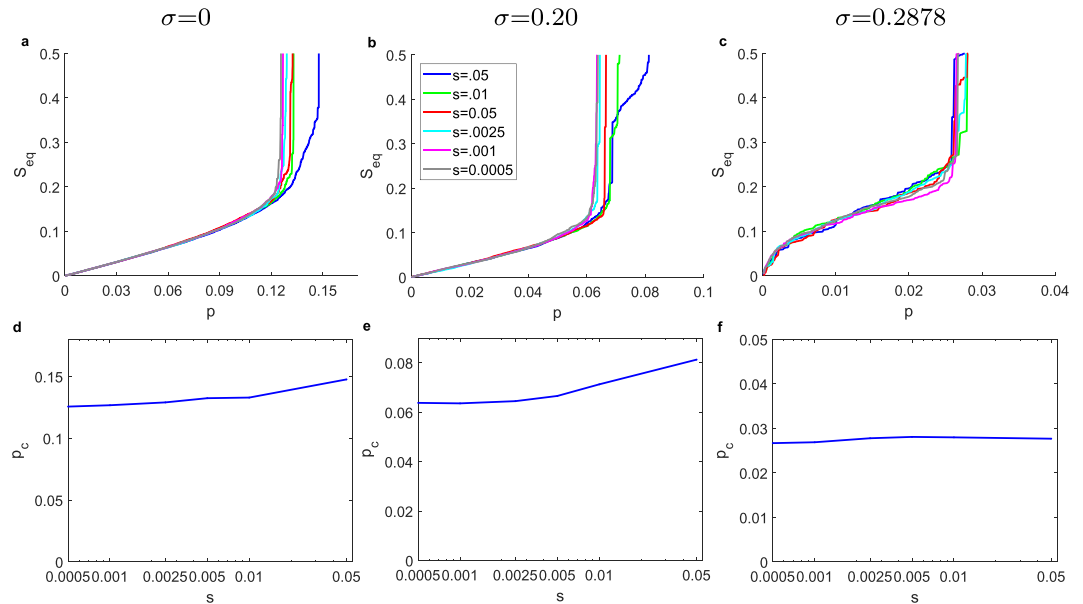


Figure 8. Impact of the size of initiator fraction s on the performance of the GPI strategy for randomizations $\nu = 25000$ for desired cascade $S_{\text{goal}} = 0.5$, for ER graphs with $N = 10,000$, $\langle k \rangle = 10$, $\rho = 0$. (**a–c**) Cascade S_{eq} vs. the initiator fraction p for $\sigma = 0, 0.2, 0.2887$ respectively (for one realization). (**e–g**) Initiator fraction s required for desired cascade $S_{\text{goal}} = 0.5$ vs. randomizations ν for $\sigma = 0, 0.2, 0.2887$ respectively (for one realization).

for very low ($\rho = -0.9$) and high ($\rho = 0.9$) degree assortativity, there are multiple initiator fractions for which a large spread occurs, which vary for the strategies, allowing for a sudden change of 'lead' between the four network depend strategies (CI-TM, ID, *deg*, and DD), with DD and ID performing the best. For larger σ the importance of resistance increases while the importance of the network structure declines, making the strategies more depended on resistance to take the 'lead', while also reducing the number of sharp 'lead' changes.

Next, we focus on the BI and GPI strategies and their performance for their different parameters. For the BI strategy, we scan the parameters space $a \times b$ and record the average minimum p_c at which the cascade size is $S_{\text{goal}} = 0.5$ (Fig. 5). The second and third feature of this weighted method defined by Eq. 1 correspond to the first two terms of the CI-TM metric. The three features are interdependent, e.g. before the cascade begins with $\sigma = 0$, a node's i resistance r_i is nearly linearly proportional to its degree k_i , which is why we observe those linear contours on the plots a, d, e. Moreover, it is clear that any strategy that would exclude the resistance ($a = 0$) from its metric, such as the CI-TM, will have inferior performance. The contours indicate that the importance of the resistance and degree of a node relative to the third feature, which in addition is computationally most costly to obtain. Furthermore, we have recorded the impact of the standard deviation σ on the optimal weights (Fig. 6). As σ increases, the optimal c coefficient decreases. Interestingly, the most important feature is the resistance ($a \approx 0.53$), then the degree ($b \approx 0.32$), and the smallest importance is left for $c \approx 0.15$. This result is especially important since other strategies do not fully utilize the resistance information combined with other network centrality measures. We further record the performance the optimal weights for BI against the degree assortativity of the networks (see Fig. 4 of the Supplementary Information).

For the GPI strategy, Figs 7 and 8 we show its performance of GPI in relationship to the number of randomizations ν and step sizes s , respectively. On average, with increasing ν and decreasing s we always minimize p_c for obtaining a cascade desired size (here we aim at the size $N/2$). On average, we expect and observe an asymptotic return with increasing ν . For computational efficiency, we fix the s and ν when the additional performance is minimal. Further investigation is required in order to find the interplay between the two control parameters in order to optimize the performance of the algorithm for the smallest computational time possible. Interestingly, for $\sigma = 0.2878$ in contrast to the direct methods, there is a large transition on cascade size as we reach S_{goal} .

Discussion

The challenge of Influence Maximization for the LTM or other diffusion processes is finding low complexity, yet well performing algorithms for the discovery of superspreaders. We tackle this challenge by introducing two different approaches. On our first approach we take advantage of the combination of different node features: the resistance to influence, the out-degree, and the impact of its activation to its second neighborhood. Those features essentially capture the resistance drop of the network. By placing weights between them and constructing the BI metric, we are able to study the importance of each of those features for a variety of network structures (degree assortativity) and threshold distributions. We discovered that resistance is often the most impactful of the features. E.g., for bidirectional graphs, the cascade size is governed not by initiators with high network centrality measures but by high resistance nodes when selected as an initiator. Moreover, a past research further demonstrates that the cascade size is governed not by superspreaders but by a critical mass of low resistance nodes

being connected to each other³⁸. These results emphasize the importance of resistance to adoption in dynamic processes.

In addition, we optimized those weights for influence maximization. Among the strategies with similar or lower complexity, BI always outperforms them for any degree assortativity or threshold distribution. Even in the case of non-optimized weights, like considering equal weights between the three features (ID strategy) or just the resistance and out-degree (DD), our method is out-performing all other strategies with similar or lower complexity ($\mathcal{O}(\langle k \rangle N + N \log N)$), which reduces to $\mathcal{O}(N \log N)$ for sparse graphs when using a sphere of influence to approximately compute the spread).

Yet, most strategies (including ours above), are based on heuristic/analytical metrics that do not take into consideration the combination of nodes, leading to local solutions of reduced size of initiators' set. Our second approach (GPI) focuses on the random combination of nodes for influence maximization, thus targeting a global optimum. It allows us to control a number of parameters which in return control the performance and computational time. The time complexity of GPI is $1/s * \mathcal{O}(\langle k \rangle N^2 + N \log N)$, which reduces to $1/s * \mathcal{O}(N \log N)$ for sparse graphs when using a sphere of influence to approximately compute the spread. Our results show that the GPI metric performs better than any strategy against we compared it for almost any initiator size, threshold distribution, and network assortativity. To say the least, GPI serves as a benchmark for (synthetic) graphs and sets a minimum bound for the optimal initiator set. Finally, in terms of applicability, both our approaches can be used for directed and weighted graphs as well, although a few adjustments would have to be made for the weighted strategy in case of weighted graphs.

Our two approaches, combining node features, and combining nodes, show there are more possible improvements to be made on both the performance and time-complexity for Influence Maximization. New methods could simultaneously take into account the network and model specific node properties as well as the combination of nodes. Other methods could possibly focus on learning to discover important node features for instance. As far as GPI is concerned, a number of improvements can be made for controlling the number of repetitions v while keeping the algorithm performance approximately the same.

Methods

For the generation of Erdős-Rényi (ER) graphs⁶³ we used the $G(N, p_{ER})$ model with N being the system size and p_{ER} the probability that a random node will be connected to any node in the graph. The probability p_{ER} is given by $p_{ER} = \langle k \rangle / (N - 1)$, where $\langle k \rangle$ is the nominal average degree in the network. The probability of the existence of a disconnected component is 4.5×10^{-5} for $\langle k \rangle = 10$ and $N = 10^4$.

The degree assortativity was first introduced by Newman⁶⁴ to describe the connectivity between neighboring nodes with different degrees. To measure it we use Spearman's ρ ⁶². ER graphs have $\rho = 0$ degree assortativity. To control the degree assortativity, we use the method applied in⁶⁵.

As far as the thresholds are concerned, they are bound between 0 and 1. The threshold distributions are truncated to generate numbers between the above two bounds³⁷. The standard deviation for a uniform threshold distribution for these bounds is $\sigma = 1/\sqrt{12} \approx 0.2887$. The truncated threshold distribution $P(\phi, \sigma)$ is given by $P(\phi, \sigma) = N(\mu, \tau) / (1 - \int_{-\infty}^0 N(\mu, \tau) d\mu - \int_1^{\infty} N(\mu, \tau) d\mu)$ for $0 \leq \phi \leq 1$, and $P(\phi, \sigma) = 0$ anywhere else, similar to³⁷. In the above, $N(\mu, \tau)$ is the normal distribution with mean μ and standard deviation τ , which take values $0 \leq \mu \leq 1$ and $0 \leq \sigma \leq \infty$ respectively.

References

- Crucitti, P., Latora, V. & Marchiori, M. A topological analysis of the Italian electric power grid. *Physica A* **338**(1-2), 92–97 (2004).
- Kinney, R., Crucitti, P., Albert, R. & Latora, V. Modeling cascading failures in the North American power grid. *Eur. Phys. J. B* **46**, 101–107 (2005).
- Hines, P., Balasubramaniam, K. & Sanchez, E. C. Cascading failures in power grids. *IEEE Potentials* **28**(5), 101–107 (2009).
- Mugisha, S. & Zhou, H. J. Identifying optimal targets of network attack by belief propagation. *Phys. Rev. E* **94**, 012305 (2016).
- Zhou, H. J. Structural resilience of directed networks. *arXiv preprint*, arXiv:1701.03404 (2016).
- Braunstein, A., Dall'Asta, L., Semerjian, G. & Zdeborová, L. Network dismantling. *Proc. Natl. Acad. Sci. USA* **113**, 12368–12373 (2016).
- Motter, A. & Lai, Y.-C. Cascade-based attacks on complex networks. *Phys. Rev. E* **66**, 065102 (2002).
- Wang, J.-W. & Rong, L.-L. Cascade-based attack vulnerability on the US power grid. *Saf Sci* **47**(10), 1332–1336 (2009).
- Asztalos, A., Sreenivasan, S., Szymanski, B. K. & Korniss, G. Cascading failures in spatially-embedded random networks. *PLOS ONE* **9**(1), e84563 (2013).
- Leskovec, J. *et al.* Cost-effective Outbreak Detection in Networks. *13th ACM SIGKDD*, 420–429 (2007).
- Kitsak, M. *et al.* Identification of influential spreaders in complex networks. *Nat. Phys.* **6**, 888–893 (2010).
- Pastor-Satorras, R., Castellano, C., Mieghem, P. V. & Vespignani, A. Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**, 925 (2015).
- Altarelli, F., Braunstein, A., Dall'Asta, L., Wakeling, J. R. & Zecchina, R. Containing epidemic outbreaks by message-passing techniques. *Phys. Rev. X* **4**, 021024 (2014).
- Malliaros, F. D., Rossi, M.-E. G. & Vazirgiannis, M. Locating influential nodes in complex networks. *Sci. Rep* **87**, 925 (2016).
- Domingos, P. & Richardson, M. Mining the network value of customers. *7th ACM SIGKDD*, 57–66 (2001).
- Richardson, M. & Domingos, P. Mining knowledge-sharing sites for viral marketing. *8th ACM SIGKDD*, 61–70 (2002).
- Barbieri, N. & Bonchi F. Influence maximization with viral product design. *SIAM ICDM*, 55–63 (2014).
- Kempe, D., Kleinberg, J. & Tardos, É. Maximizing the spread of influence through a social network. *9th ACM SIGKDD*, 137–146 (2003).
- Chen, W., Yuan, Y. & Zhang, L. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. *IEEE ICDM*, 88–97 (2010).
- Goyal, A., Lu, W. & Lakshamanan, L. V. S. CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. *ACM WWW*, (Companion Volume) (2011).
- Goyal, A., Lu, W. & Lakshamanan, L. V. S. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. *11th IEEE ICDM*, 211–220 (2011).

22. Wang, C., Chen, W. & Wang, Y. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Min. Knowl. Disc.* **25**(3), 545–576 (2012).
23. Jankowski, J. *et al.* Sequential Seeding in Complex Networks: Trading Speed for Coverage. *Sci. Rep.* **7**, 891 (2017).
24. Jiang, Q. *et al.* Simulated Annealing Based Influence Maximization in Social Networks. *25th AAAI*, 127–132 (2011).
25. Lü, L. *et al.* Vital nodes identification in complex networks. *Physics Reports* **650**, 1–63 (2016).
26. Shakarian, P., Eyre, S. & Paulo, D. A scalable heuristic for viral marketing under the tipping model. *Soc. Netw. Anal. Min.* **3**, 1225–1248 (2013).
27. Morone, F. & Makse, H. A. Influence maximization in complex networks through optimal percolation. *Nature* **524**, 65–68 (2015).
28. Pei, S., Teng, X., Shaman, J., Morone, F. & Makse, H. Efficient collective influence maximization in threshold models of behavior cascading with first-order transitions. *Sci. Rep.* **7**, 45240 (2017).
29. Granovetter, M. Threshold Models of Collective Behavior. *Am. J. Sociol.* **83**, 1420–1443 (1978).
30. Schelling, T. C. Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *J. Confl. Resolut.* **17**, 381–428 (1973).
31. Baxter, G. J., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. F. Bootstrap percolation on complex networks. *Phys. Rev. E* **82**, 011103 (2010).
32. Baxter, G. J., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. F. Heterogeneous k-core versus bootstrap percolation on complex networks. *Phys. Rev. E* **83**, 051134 (2011).
33. Amini, H. Bootstrap percolation in living neural networks. *J. Stat. Phys.* **141**, 459–475 (2010).
34. Eckmanna, J.-P. *et al.* The physics of living neural networks. *Physics Reports* **449**(1–3), 54–76 (2007).
35. Kozma, R., Puljic, M., Balister, P., Bela Bollobás, B. & Freeman, W. J. Phase transitions in the neuropercolation model of neural populations with mixed local and non-local interactions. *Biol. Cybern.* **92**, 367–379 (2005).
36. Lympereopoulos, I. N. & Ioannou, G. D. Online social contagion modeling through the dynamics of Integrate-and-Fire neurons. *Inf. Sc.* **320**, 26–61 (2015).
37. Karampourniotis, P. D., Sreenivasan, S., Szymanski, B. K. & Korniss, G. The Impact of Heterogeneous Thresholds on Social Contagion with Multiple Initiators. *PLoS One* **10**(11), e0143020 (2015).
38. Watts, D. J. & Dodds, P. S. Influential networks and public opinion formation. *J. Cons. Res.* **34**, 441–458 (2007).
39. Singh, P., Sreenivasan, S., Szymanski, B. K. & Korniss, G. Threshold-limited spreading in social networks with multiple initiators. *Sci. Rep.* **3**, 2330 (2013).
40. Gleeson, J. P. Cascades on correlated and modular random networks. *Phys. Rev. E* **77**, 046117 (2008).
41. Ikeda, Y., Hasegawa, T. & Nemoto, K. Cascade dynamics on clustered network. *J. Phys.: Conf. Ser.* **221**, 012005 (2010).
42. Lee, K.-M., Brummitt, C. D. & Goh, K.-I. Threshold cascades with response heterogeneity in multiplex networks. *Phys. Rev. E* **90**, 062816 (2014).
43. Nematzadeh, A., Ferrara, E., Flammini, A. & Ahn, Y.-Y. Optimal network modularity for information diffusion. *Phys. Rev. Lett.* **113**, 088701 (2014).
44. Curato G. & Lillo, F. Optimal information diffusion in stochastic block models. *Phys. Rev. E* **94**(3) (2016).
45. Latané, B. & L'Herrou, T. Spatial Clustering in the Conformity Game: Dynamic Social Impact in Electronic Groups. *J. Pers. Soc. Psychol.* **70**, 1218–1230 (1996).
46. Centola, D. The Spread of Behavior in an Online Social Network Experiment. *Science* **329**, 5996 (2010).
47. Monsted, B., Sapiezynski, P., Ferrara E. & Lehmann S. Evidence of Complex Contagion of Information in Social Media: An Experiment Using Twitter Bots *arXiv preprint*, arXiv:1703.06027 (2017).
48. Romero, D. M., Meeder, B. & Kleinberg J. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In Srinivasan, S. *et al.* (eds) *ACM WWW* **20**, 695–704 (2011).
49. Karsai, M., Iniguez, G., Kikas, R., Kaski, K. & Kertész, J. Local cascades induced global contagion: How heterogeneous thresholds, exogenous effects, and unconcerned behaviour govern online adoption spreading. *Sci. Rep.* **6**, 27178 (2016).
50. Fink, C., Schmidt, A., Barash, V., Cameron, C. & Macy, M. Complex contagions and the diffusion of popular Twitter hashtags in Nigeria. *Soc. Netw. Anal. Min.* **6**(1) (2016)
51. Fink C. *et al.* Investigating the Observability of Complex Contagion in Empirical Social Networks. *10th AAAI*, 121–130 (2016)
52. State, D. & Adamic, L. *The Diffusion of Support in an Online Social Movement: Evidence from the Adoption of Equal-Sign Profile Pictures, Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 1741–1750 (ACM, New York, NY, 2015).
53. Unicomb, S., Iniguez, G. & Karsai, M. Threshold driven contagion on weighted networks. *Sci. Rep.* **8**, 3094 (2018).
54. Tang, Y., Xiao, X. & Shi, Y. Influence Maximization: Near-Optimal Time Complexity Meets Practical Efficiency. *ACM SIGMOD'15*, 75–86 (2014).
55. Liu, X. *et al.* IMGPU: GPU-Accelerated Influence Maximization in Large-Scale Social Networks. *IEEE Trans. Parallel. Distrib. Syst.* **25**(1), 136–145 (2014).
56. Lu, Z., Zhang, W., Wu, W., Kim, J. & Fu, B. The complexity of influence maximization problem in the deterministic linear threshold model. *J. Comb. Optim.* **24**, 374–378 (2012).
57. Watts, D. J. A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci. USA* **99**, 5766–5771 (2002).
58. Gleeson, J. P. & Cahalane, D. J. Seed size strongly affects cascades on random networks. *Phys. Rev. E* **75**, 056103 (2007).
59. Altarelli, F., Braunstein, A., Dall'Asta, L. & Zecchina, R. Optimizing spread dynamics on graphs by message passing. *J. Stat. Mech.* **9**, P09011 (2013).
60. Lim, Y., Ozdaglary, A. & Teytelboymz, A. A Simple Model of Cascades in Networks (2015).
61. Weskida, M. & Michalski, R. Evolutionary Algorithm for Seed Selection in Social Influence Process. *IEEE/ACM ASONAM* **17**, 1189–1196 (2016).
62. Hovstad, R. & Litvak, N. Degree-Degree Dependencies in Random Graphs with Heavy-Tailed Degrees. *Internet Math* **10**, 287–334 (2014).
63. Erdős, P. & Rényi, A. On random graphs. *Publ. Math. Debrecen* **6**, 290–297 (1959).
64. Newman, M. E. J. Assortative mixing in networks. *Phys. Rev. Lett.* **89**, 208701 (2002).
65. Molnár, F. Jr. *et al.* Dominating Scale-Free Networks Using Generalized Probabilistic Methods. *Sci. Rep.* **4**, 6308 (2014).

Acknowledgements

The authors thank Prof. Radoslaw Michalski for his comments on this work. This work was supported in part by the Army Research Office grant W911NF-12-1-0546, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by the Office of Naval Research Grant No. N00014-09-1-0607 and N00014-15-1-2640. B.K.S. and G.K. were also supported by the Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) under Contract No. W911NF-17-C-0099, and B.K. was also supported by the National Science Centre, Poland, project no. 2016/21/B/ST6/01463. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied of the Army Research Laboratory or the U.S. Government.

Author Contributions

P.D.K., B.K.S. and G.K. designed the research; P.D.K. implemented and performed numerical experiments and simulations; P.D.K., B.K.S. and G.K. analyzed data and discussed results; P.D.K., B.K.S. and G.K. wrote, reviewed, and revised the manuscript.

Additional Information

Supplementary information accompanies this paper at <https://doi.org/10.1038/s41598-019-41822-w>.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019