Evaluate the Robustness of Influence Maximization Against Edge Uncertainty

Yikai Yan 517021910404 [1]

Abstract

Influence Maximization (IM for short) has been intensively studied in the past two decades. Before running IM algorithms, we should first model the real social network, which may be not accurate enough. Therefore, evaluating the robustness of IM strategies when the edge values have error is important. We give a reasonable formulation of the robustness evaluation problem against edge uncertainty. Furthermore, we study the discrete and continuous case of the problem, about the NP-hardness, inapproximability, and submodularity of the problem. For the discrete case, we propose a greedy algorithm with $1 - 1/e$ approximation ratio under the linear threshold model. For the continuous case, we propose a SGD algorithm. We implement and evaluate the two algorithms over a simulated social network. The evaluation results demonstrate that the proposed greedy algorithm outperforms the naive baseline, and validate the feasibility of the SGD algorithm.

1. Introduction

Influence Maximization was first proposed by Kempe et al., 2003, and has been intensively studied in the past two decades (Jiang et al., 2011; Song et al., 2015; Li et al., 2017; Kalimeris et al., 2019; Chen et al., 2020). The seminal work proposed two classic influence diffusion models: Linear Threshold Model (LTM) and Independent Cascade Model (ICM), based on which they further proposed a greedy algorithm to solve the IM problem, and proved an $1 - 1/e$ approximation ratio for the algorithm by the submodularity theory (Nemhauser et al., 1978).

Before solving the IM problem, we have to first model the real social network, using LTM, ICM, or other diffusion models. Accurately modeling the real social network significantly contributes to good performance for the IM strategy. Otherwise, even if the IM problem is solved perfectly, the picked seeds may not work well on the real social network. Therefore, much efforts have been expended in how to model the real social network, including last year's course project (Jiang & Zhao, 2019), which models the social network based on community.

However, it is hard to accurately model a social network. On one hand, edges registered in the system may be missing in the real social network. On the other hand, edge values of the real social network may be affected by a number of unmodeled factors, and may deviate from the modeled values. Therefore, classic IM algorithms that are completely based on the modeled social network may suffer from poor robustness, but only little work has studied the robustness problem (Chen et al., 2016; Bogunovic et al., 2017). The work (Bogunovic et al., 2017) studied the robust submodular function optimization with a $\max$-$\min$ objective. However, they only studied the robustness against node uncertainty. Since the objective is a function on the node set ($\sigma(V)$), studying the robustness against the uncertainty of the independent variable V is essentially different from studying that of the parameters of the function (edge values $\mathbf{E}$). The work (Chen et al., 2016) studied the robustness against edge uncertainty, but made a very unrealistic assumption: each edge value has an **independent** range, that is, $\mathbf{E}$ is assumed to be within a "rectangular" range. As mentioned earlier, edge values are affected by a number of unmodeled factors, and are thus more reasonable to be assumed as following the joint Gaussian distribution. The key difference is that under the Gaussian distribution assumption, the edge values $\mathbf{E}$ is within a "sphere" range, which is much more difficult to study than a "rectangular" range. Under their assumption, we can simply regard the combination of lowest values of each independent range as the worst case. However, under the practical Gaussian distribution assumption, we have to evaluate each edge value's impact on the diffusion process.

To establish robust IM algorithms against edge uncertainty, we study one of the main steps of it: evaluating the robustness of a given IM seed set. The problem is, given a social graph and a seed set, querying the minimum influence expectation of the seed set. In addition to laying the foundation for a robust algorithms, solving this problem can also be used for risk assessment of IM strategies. We give two formulations of this problem, deviding it into discrete and continuous cases. For the discrete case where there are at most l edges missing, we propose a greedy algorithm, and

[1]Shanghai Jiao Tong University, China. Correspondence to: Yikai Yan <1062352133@qq.com>.

Project of EE447-1. Theme: Influence Maximization.

Table 1. Theoretical results of the greedy algorithm for the robustness evaluation problem (discrete) under different diffusion models.

Diffusion Model	NP-hardness	Submodularity
Linear Threshold	Uncertain	Satisfied
Deterministic Linear Threshold	Proved	Not Satisfied
Independent Cascase	Proved (Inapproximability also Proved)	Not Satisfied

study the NP-hardness, inapproximability, and submodularity of the problem. The theoretical results are summarized in Table 1. For the continuous case where the edge values $\mathbf{E}$ follow a joint Gaussian distribution, we propose a SGD algorithm to optimize the objective function. We summarize the contributions of this work as follows.

- To the best of our knowledge, we are the first to study the robustness of IM strategies against edge uncertainty, and give a reasonable formulation of it.
- We thoroughly study the discrete case of the robustness evaluation problem, and propose a greedy algorithm to solve it. We prove the NP hardness, inapproximability, and the submodularity (or non-submodularity), under various diffusion models.
- We also propose a SGD algorithm to solve the continuous case of the robustness evaluation problem.
- We implement the proposed algorithms and evaluate the performance. The evaluation results validate the theoretical results and the feasibility of our algorithms, and demonstrate that the proposed greedy algorithm outperforms the naive baseline.

2. Problem Formulation

We formulate the problem of evaluating the robustness of an IM strategy as the following *Influence Minimization* problem.

$$RI(G,\hat{V}) = \min_{\hat{\mathbf{E}}\in\mathcal{D}_{\mathbf{E}}} \sigma((V,\hat{\mathbf{E}}),\hat{V}), \quad G=(V,\mathbf{E}). \tag{1}$$

The graph $G=(V,\mathbf{E})$ represents the social network which is based on when selecting the optimal seeds in the IM process. $\hat{V}$ is the seeds picked in the IM process, and $RI(G,\hat{V})$ means the robust influence of this IM strategy. We use $\mathcal{D}_{\mathbf{E}}$ to denote the possible value range of the real social network's edges, and σ is a function which outputs the expectation influence of seeds $\hat{V}$ on the graph $(V,\hat{\mathbf{E}})$.

Although this work focuses on evaluating the robustness of an IM strategy, it may inspire studies on the following Robust IM (RIM for short) problem against edge uncertainty.

$$RIM(G) = \max_{\hat{V}\subseteq V, |\hat{V}|=k} RI(G,\hat{V}) \tag{2}$$

In the RIM process, the decision maker has only access to the estimated social network G, and she wants to maximize the worst case influence expectation when the edges of the real network deviates from $\mathbf{E}$.

For the propagation models for the objective function σ, we dicuss three models in this work: Independent Cascade Model (ICM), Linear Threshold Model (LTM), and Deterministic Linear Threshold Model (DLTM). The first two models are the classical ones from the seminal work (Kempe et al., 2003). In ICM, $\mathbf{E}$ is a matrix of probabilities. When the node u is activated, it tries to activate each of its neighbors v with probability E_{uv} only once. In LTM, $\mathbf{E}$ is a matrix of edge weights, and E_{uv} is the weight of edge (u,v), subject to that $\sum_u E_{uv} \le 1,\ \forall v$. Edge v is activated iff $\sum_{u \text{ active neighbor of } v} E_{uv} \ge \theta_v$. θ_v is the threshold of node v, which is picked uniformly at random from $[0,1]$. We extraly discuss the DLTM, where θ_v is a certain value determined according to the social network rather than a random value. DLTM is more reasonable because it enables the decision maker to take into account the prior knowledge about how easily each node can be activated. However, it's hard to obtain theoretical performance guarantee for algorithms in this model, because the submodularity is often not satisfied. In the following two subsections, we give two specific formulation in the discrete case and the continuous case based on (1).

2.1. Discrete Case

In the discrete case, we focus on the following problem that queries the minimum influence when at most l edges in E[1] actually do not exist.

$$\max_{\hat{E}\subseteq E, |E-\hat{E}|\le l} \Big(\sigma((V,\mathbf{E}),\hat{V}) - \sigma((V,\hat{\mathbf{E}}),\hat{V})\Big), \tag{3}$$

where we transform the $\min$ problem to an equivalent $\max$ problem. The intuitive meaning of (3) is to query the maximum negative influence to a given IM strategy of removing l edges from E. The correspondence with (1) is $\mathcal{D}_{\mathbf{E}} = \{\hat{E} \mid \hat{\mathbf{E}} \le \mathbf{E} \text{ and } \|\text{vec}(\mathbf{E}-\hat{\mathbf{E}})\|_0 \le l\}$, where $\hat{\mathbf{E}} \le \mathbf{E}$ means each element satisfies this inequation and

[1] E is the edge set corresponding to matrix $\mathbf{E}$, and $\hat{E}$ is the one for matrix $\hat{\mathbf{E}}$

$\|\cdot\|_0$ is the L_0 norm[2]. The intuition behind this formulation is that users may have friendship links in the system, but have no influence on each other. For example, a student registers friendship relation with her teacher in Wechat, but they never contact each other once the course is completed. We term this case the Discrete Robustness Evaluation problem under IC/LT/DLT model (**DER-IC/LT/DLT** for short).

2.2. Continuous Case

We first formulate the continuous case according to (1).

$$\min_{\|\hat{\mathbf{E}}'-\mathbf{E}'\|_2 \le R} \sigma((V,\hat{\mathbf{E}}),\hat{V}), \tag{4}$$

where $\mathbf{E}'$ is the flattened vector of matrix $\mathbf{E}$, discarding elements with value 0 (0 means that this edge does not exist). For $\hat{\mathbf{E}}$, we add the constraint that if an edge does not exist in $\mathbf{E}$, it does not exist in $\hat{\mathbf{E}}$ either, and use $\hat{\mathbf{E}}'$ to denote the vector of elements corresponding to those in $\mathbf{E}'$.

The decision maker often estimates the edge values by some particular model. However, the real edge value may be affected by many unmodeled factors. Hence, it is natural to assume that the real edge values $\hat{\mathbf{E}}'$ follows the Gaussian distribution $N(\mathbf{E}',\sigma^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Under this assumption, the probability density of $\hat{\mathbf{E}}'$ is $\Pr(\hat{\mathbf{E}}') = 1/(2\pi\sigma^2)^{|\hat{\mathbf{E}}'|/2}\exp(-\|\hat{\mathbf{E}}'-\mathbf{E}'\|_2^2/2\sigma^2)$. We consider $\hat{\mathbf{E}}'$ with probability density under a certain threshold as impossible, then the possible value range $\mathcal{D}_{\mathbf{E}}$ is a sphere space $\{\hat{\mathbf{E}}' \mid \|\hat{\mathbf{E}}'-\mathbf{E}'\|_2 \le R\}$, where the radius R is associated with σ and the threshold.

The minimization problem under the L_2-norm constraint is hard to optimize, and we relax the problem into an unconstained minimization problem with an L_2 regularization term.

$$\min_{\hat{\mathbf{E}} \text{ has only edges in } E} \sigma((V,\hat{\mathbf{E}}),\hat{V}) + \lambda \left\|\hat{\mathbf{E}}'-\mathbf{E}'\right\|_2^2, \tag{5}$$

where λ is negatively correlated with the edge uncertainty. We term this case continuous robustness evaluation problem under IC/LT/DLT model (**CER-IC/LT/DLT** for short).

3. Complexity Analysis

In this section, we analyze the complexity of the robustness evaluation problem in the discrete case. We prove DER-DLT and DER-IC are both NP-hard. In addition, we prove the inapproximability of DER-IC.

Theorem 1. *DER-DLT is NP-hard.*

Proof of Theorem 1. We prove the NP-hardness by the reduction from a maximum set union problem[3], defined by a collection of subsets $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ of a ground set $U = \{u_1, u_2, \cdots, u_n\}$. This problem queries whether we can select exactly k subsets $S_{j_1}, \cdots, S_{j_k}$ from $\mathcal{S}$ whose union size $|S_{j_1} \cup \cdots \cup S_{j_k}| = l$. The well-known NP-hard set cover problem is a sub-problem of the maximum set union problem, where $l = n$. Hence, this problem is also NP-hard, and we only need to show the reduction from this problem to prove the NP-hardness of DER-DLT.

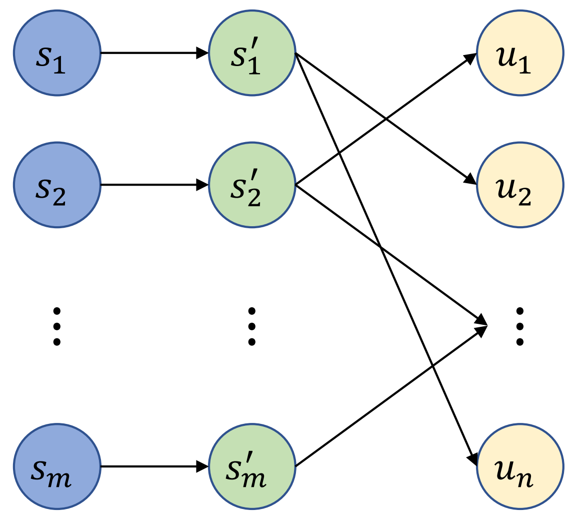


Figure 1. The discrete robustness evaluation problem that is reduced from the maximum set union or maximum set intersection problem

With Figure 1, we show the reduced DER-DLT problem. G is difined by the figure, where the thresholds for each node are all 1. $Es_i, s'_i = 1, \forall i \in [m]$[4], and $E_{s'_i,u_j} = 1/d_{in}(u_j)$. $d_{in}(u_j)$ is the in degree of u_j, we set $E_{s'_i,u_j}$ like this to ensure that each u_j is activated iff all nodes connected to it are activated. Note that s_i and s'_i correspond to set S_i, and u_i corresponds to element u_i. s'_i is connected with u_j iff S_i contains u_j. The initial seeds of the IM strategy is $\hat{V} = \{s_1, s_2, \cdots, s_m\}$, which has influence $2m + n$ (activates all nodes). Then, the DER-DLT problem queries whether we can reduce $k + l$ activated nodes if we remove k edges from G.

Then, we show the equivalence. First, note that we only need to consider removing edges from s_i to s'_i and do not consider those from s'_i to u_j. This is because if we remove edge (s'_i, u_j), removing (s_i, s'_i) instead can deactivate no less nodes (if we remove (s_i, s'_i), edges

[2] L_0 norm is the number of non-zero elements

[3] We discuss the decision versions rather than the optimization ones of these two problems for conciseness. Similarly, we do so for the DER-IC as well.

[4] $[m]$ is the set $\{1, 2, \cdots, m\}$

from s'_i, including (s'_i, u_j) will have no effect). Second, choosing sets $S_{j_1}, \cdots, S_{j_k}$ corresponds to choosing edges $(s_{j_1}, s'_{j_1}), \cdots, (s_{j_k}, s'_{j_k})$. When the union size is l', the number of deactivated nodes is $k + l'$, and vise versa. Therefore, the equivalence is built, and we can get an l-size union set iff we can deactivate $k + l$ nodes. □

Theorem 2. *MER-IC is both NP-hard and inapproximable*

1. *MER-IC is NP-hard.*
2. *Let $\varepsilon > 0$ be an arbitrarily small constant. Assume that SAT does not have a probabilistic algorithm that decides whether a given instance of size n is satisfiable in time 2^{n^ε}, which is a standard assumption. Then there is no polynomial time algorithm for MER-IC-modified (trivial modification defined later in the proof) that achieves an approximation ratio of $1/(N-2n)^{\varepsilon'}$ where N is the size of the instance, and ε' depends only on ε.*

Proof of Theorem 2. We first introduce a lemma proved by Xavier, 2012 about the maximum set intersection problem. This problem is the one replacing the union set in the maximum set union problem by the intersection set.

Lemma 1. *The maximum set intersection problem is both NP-hard and inapproximate. The inapproximability result is in the same form with that of Theorem 2 (the inapproximation ratio for this problem is $1/N^{\varepsilon'}$).*

With this lemma, we only need to prove the reduction from the maximum set intersection problem to prove Theorem 2. The MER-IC-modified problem is to to maximize $\sigma((V, \mathbf{E}), \hat{V}) - \sigma((V, \hat{\mathbf{E}}), \hat{V}) - k$. We minus k to make the objective function value same as that of the maximum set intersection problem.

The reduced DER-IC-modified problem is also shown by Figure 1. The only difference is that each edge has probability 1, such that u_j is activated if any one node connected to it is activated. Then the reduced DER-IC-modified problem is querying whether we can remove k edges to deactivate $k + l$ nodes (achieve l in the objective function). Since we need to deactivate all edges connected to u_j to deactivate u_j, this DER-IC-modified problem is equivalent with the maximum set intersection problem. Hence, the NP-hardness and of DER-IC are proved. For the inapproximation result, we note that the constructed MER-IC-modified has $2n$ larger instance size than the maximum set intersection problem (n nodes and edges are added), and the objective function value is the same. Therefore, the $1/(N-2n)^{\varepsilon'}$ inapproximability result is proved. □

The NP-hardness of DER-LT is reserved in the future work, and the continuous case does not have natural NP-hardness formulation, thus not considered. Due to the NP-hardness of DER-DLT and DER-IC, we can not expect a polynomial algorithm for these two problems, and due to the inapproximability of DER-IC, we can not expect a $1 - 1/e$ approximation ratio of greedy algorithms for DER-IC.

Algorithm 1 Greedy Algorithm for DER-LT/DLT/IC

1: **Input:** graph $G = (V, \mathbf{E})$; seeds $\hat{V} \subseteq V$; the maximum number of edges to be removed l; influence propagation model σ (may be DLT/LT/IC); number of simulations to calculate the influence expectation n_{test}
2: Do initialization:
$$\bar{E} \leftarrow \emptyset.$$
3: **while** $|\bar{E}| < l$ **do**
4: Query the edge to maximize the reduced influence (simulate n_{test} times to calculate σ for LT and IC model):
$$\hat{e} \leftarrow \arg\max_{e \in E - \bar{E}} \Big(\sigma((V, E), \hat{V}) - \sigma((V, E - \bar{E} - \{e\}), \hat{V})\Big).$$
5: $\bar{E} \leftarrow \bar{E} \cup \{\hat{e}\}$.
6: **end while**
7: **Return:** edges $\bar{E}$ and the corresponding influence expectation $\sigma((V, E - \bar{E}), \hat{V})$.

4. Discrete Formulation: Greedy Algorithm

In this section, we present a greedy algorithm for DER-LT/DLT/IC, and prove an $1 - 1/e$ approximation ratio for DER-LT. We also show that the objective functions for DER-DLT/IC are not submodular, and some possible modification to these two cases to achieve a theoretical approximation ratio is reserved in the future work.

We present the algorithm in Algorithm 1. The process is to greedily maximize the objective function (reduced influence) $f_D(\bar{E}) = \sigma((V, E), \hat{V}) - \sigma((V, E - \bar{E}), \hat{V})$, by iteratively adding edges into $\bar{E}$. Greedy algorithms in the IM background usually have high time complexity and are hard to scale up (Cohen et al., 2014). Unfortunately, Algorithm 1 also suffers from this problem, even more severely, since it iterates over edges rather than nodes. Specifically, by simple calculation, the time complexity is in the $O(l n_{test} |E|^2)$ order (suppose $|E| \geq |V|$).

Then, we prove the submodularity when the diffusion model is LT.

Theorem 3. *The objective function $f_D(\bar{E})$ under the LT diffusion model is submodular. That is,*

$$f_D(E_1 \cup \{e\}) - f_D(E_1) \geq f_D(E_2 \cup \{e\}) - f_D(E_2),$$

for all edges e and all pairs of sets $E_1 \subseteq E_2$.

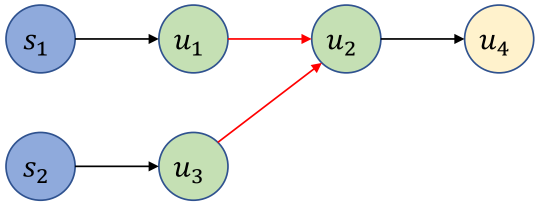


Figure 2. The live-edge graph when there are two paths from the seeds to u_4

Proof of Theorem 3. Following the tradition of the seminal work (Kempe et al., 2003), we define the live-edge graph for LT model. Each node v_j picks at most one of its incoming edges, selecting the edge from v_i with probability E_{v_i,v_j}. Then the picked edges and V forms a live-edge graph. The seminal work (Kempe et al., 2003) has proved that the expectation influence is the expectation of reachable nodes in the live-edge graph. Then, we only need to prove that for each live-edge graph, removing edge e can not deactivate more edges when more edges have been removed before e.

Based on Figure 2, we prove the following lemma.

Lemma 2. *For each node on an arbitrary live-edge graph, there is at most one path from the seeds.*

Suppose there are more than one paths to node u_4, we can pick two of them. Then, the two paths will form a graph like Figure 2. There must be a node which has two incoming edges (node u_2 in the figure), which contradicts the construction of live-edge graphs (each node has at most one incoming edges).

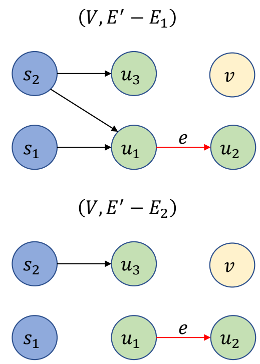
With Lemma 2, we can prove the theorem. First, we define $A(v, G)$, which is 1 when v is reachable from the seeds in live-edge graph G, and is 0 else. Then, we want to prove that for all node v, and any live-edge graph $G' = (V, E')$,

$$\begin{aligned} &A(v,(V,E'-E_1)) - A(v,(V,E'-E_1-\{e\})) \\ \geq &A(v,(V,E'-E_2)) - A(v,(V,E'-E_2-\{e\})), \end{aligned}$$

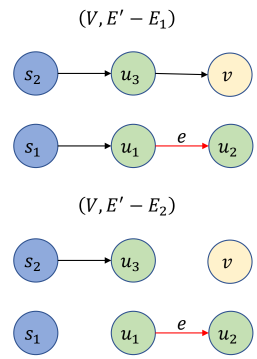
which implies the effect of removing e on v. With Figure 3, we prove this equation case by case. In case 3(a), removing e has no effect on v, and

$$\begin{aligned} &A(v,(V,E'-E_1)) - A(v,(V,E'-E_1-\{e\})) \\ = &A(v,(V,E'-E_2)) - A(v,(V,E'-E_2-\{e\})) = 0. \end{aligned} \tag{6}$$

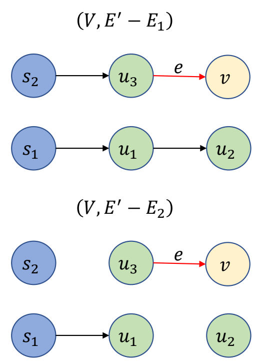
In case 3(b) and 3(c), v is reachable. From Lemma 2, there is only one path, thus following the figures. In case 3(b), removing e also does not have effect on v, and (6) still holds. In case 3(c), removing v from $(V, E' - E_1)$ can deactivate



(a) v is not reachable from any seed.



(b) v is reachable, but e is not in the path



(c) v is reachable, and e is in the path

Figure 3. The listed cases of the live-edge graph. s_1 and s_2 are selected as seeds.

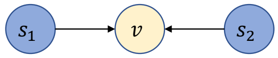


Figure 4. The example DER problem that does not satisfy submodularity. s_1 and s_2 are the seeds.

v, and

$$\begin{aligned}&A(v,(V,E'-E_1))-A(v,(V,E'-E_1-\{e\}))=1\\ \geq &A(v,(V,E'-E_2))-A(v,(V,E'-E_2-\{e\})).\end{aligned}$$

Therefore,

$$\begin{aligned}&f_D(E_1\cup\{e\})-f_D(E_1)\\ =&\mathbb{E}\sum_{v\in V}\left(A(v,(V,E'-E_1))-A(v,(V,E'-E_1-\{e\}))\right)\\ \geq&\mathbb{E}\sum_{v\in V}\left(A(v,(V,E'-E_2))-A(v,(V,E'-E_2-\{e\}))\right)\\ =&f_D(E_2\cup\{e\})-f_D(E_2)\end{aligned}$$

□

From Theorem 3 and the theory of submodular functions (Nemhauser et al., 1978), we prove an $1-1/e$ approximation ratio for Algorithm 1 under DER-LT.

Theorem 2 implies that the objective under DER-IC can not be submodular. Otherwise, the polynomial-time greedy algorithm can achieve $1-1/e$ approximation ratio. Actually, we can give simple examples to show that both the objective functions under DER-IC and DER-DLT are not submodular. The example is shown in Figure 4. For DER-IC, all the edges are with probability 1. Then,

$$\begin{aligned}&f_D(\emptyset\cup\{(s_1,v)\})-f_D(\emptyset)=0\\ <&f_D(\{(s_2,v)\}\cup\{(s_1,v)\})-f_D(\{(s_2,v)\})=1.\end{aligned}\tag{7}$$

For DER-DLT, all the edges are with weight 0.5, and the threshold for v is 0.5. Then (7) still holds. Therefore, f_D under DER-IC and DER-DLT is not submodular, and we can not obtain an approximation ratio with the theory of submodular functions.

5. Continuous Formulation: SGD

Referring to (5), we want to minimize

$$f_C(\hat{\mathbf{E}})=\sigma((V,\hat{\mathbf{E}}),\hat{V})+\lambda\left\|\hat{\mathbf{E}}'-\mathbf{E}'\right\|_2^2\tag{8}$$

in the coninuous case. Since the domain is continuous, we consider applying gradient descent on the objective function f_C to optimize it. However, we can not algibraically derive the gradient of f_C, and use simulations to estimate the gradient instead (numerical gradient). This introduces randomness to the algorithm, and we thus term the algorithm stochastic gradient descent (SGD for short). Note that this SGD is not the same as that in machine learning. We do not consider $CER-DLT$ in this work, because f_C under DLT diffusion model is not continuous or derivable.

Algorithm 2 SGD Algorithm for CER-LT/IC

Input: graph $G=(V,\mathbf{E})$; seeds $\hat{V}\subseteq V$; regularization weight λ; objective function f_C; number of simulations to calculate the gradients n_{test}; perturbation to the edge values when calculating the gradients δ; step size η; number of iteration n_{it}

Do initialization:
$$\hat{\mathbf{E}}\leftarrow\mathbf{E}.$$

for n_{it} times **do**

  Simulate n_{test} times to derive the original objective in this iteration: $f_C(\hat{\mathbf{E}})$.

  for each edge $E_{v_i,v_j}\in E$ **do**

    Perturb the value of this edge
$$\hat{\mathbf{E}}^{tmp}\leftarrow\hat{\mathbf{E}},\ \hat{E}^{tmp}_{ij}\leftarrow\hat{E}_{ij}+\delta.\tag{9}$$

    Simulate n_{test} times to derive the new objective value: $f_C(\hat{\mathbf{E}}^{tmp})$.

    Derive the numerical gradient:
$$\mathbf{g}_{ij}\leftarrow\max(\frac{f_C(\hat{\mathbf{E}}^{tmp})-f_C(\hat{\mathbf{E}})}{\delta},0)\tag{10}$$

    Add the gradient of the regularization term:
$$\mathbf{g}_{ij}\leftarrow\mathbf{g}_{ij}+2\lambda\left(\hat{E}_{ij}-E_{ij}\right)\tag{11}$$

  end for

  Update the edge values: $\hat{\mathbf{E}}_{ij}\leftarrow\hat{\mathbf{E}}_{ij}-\eta\mathbf{g}_{ij}$ for each edge $E_{v_i,v_j}\in E$.

end for

Return: edge values $\hat{\mathbf{E}}$ and the corresponding influence expectation $\sigma((V,\hat{\mathbf{E}}),\hat{V})$.

We present the details in Algorithm 2, where we apply gradient descent on the edge values $\hat{\mathbf{E}}$. The inner for loop describes the process to derive the numerical gradient. For each edge value in $\hat{\mathbf{E}}$, we purturb this value by δ[5]. Then in (10), we use the difference of influence expectation to derive the numerical partial derivative to this edge value. Since the gradient is calculated via simulation, it may be smaller than 0. However, from the practical meaning of f_C, the partial derivative can not be smaller than 0, and we apply the function $\max(\cdot,0)$ in (10). For the gradient of

[5]We do not allow $\hat{E}^{tmp}_{ij}$ to exceed 1, and let δ be the actual added value in this case.

the quadratic regularization term, we can directly derive the algebraical gradient in (11). Finally, we update $\hat{\mathbf{E}}$ and turn to the next iteration.

Similar to the greedy algorithm in the discrete case, since we also need to simulate in the SGD algorithm, the time complexity is still very high. Specifically, the time complexity is in the $O(n_{it}n_{test}|E|^2)$ order (suppose $|E| \geq |V|$).

6. Experimental Results

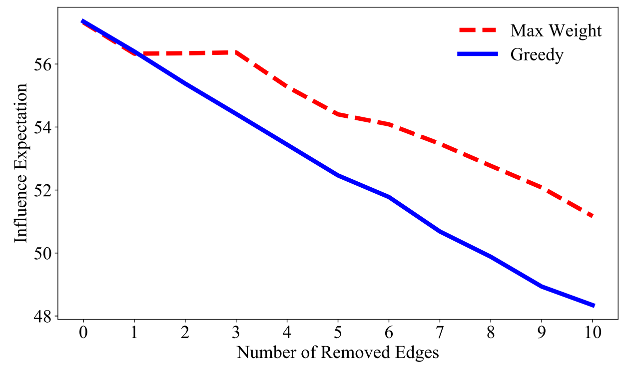
In this section, we evaluate greedy Algorithm 1 and SGD Algorithm 2 over a toy dataset under LT and IC diffusion models. As has been mentioned in earlier sections, the proposed algorithms have high time complexity. Due to the limitation of our computation resources, we can only run the algorithms on a graph with about 100 nodes. Therefore, we have to construct a tiny social network for our simulation. Specifically, we select one of the ego networks from the facebook network (McAuley & Leskovec, 2012), and randomly pick 88 nodes from it. The facebook network is undirected, and we make it directed simply by randomly assign each edge a direction. We use the contructed graph for our experiments, which contains 88 nodes and 452 edges. For the probability associating with the edges in the IC model, we follow the tradition of the seminal work (Kempe et al., 2003), assigning the probability of edge (v_i, v_j) to $1/d_{in}(v_j)$. Similarly, we assign the weight of edge (v_i, v_j) to $1/d_{in}(v_j)$ in the LT model.

6.1. Experiments on DER

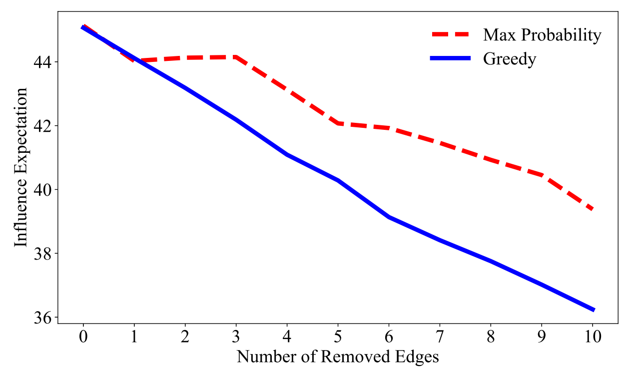
For all experiments in the DER case, we set the maximum number of edges to be removed $l = 10$. Since we are the first to study the DER problem, there is no baseline algorithm for this problem. Therefore, we use a naive algorithm, which picks the edge with the largest probability/weight in each iteration, as the baseline. We term this baseline algorithm max probability/weight.

Figure 5 shows the process of DER-LT and DER-IC. Note that the objective is to maximize the reduced influence expectation. Then we can see that the proposed greedy algorithm outperforms the baseline in both cases. Greedy algorithm reduces 2.653 more influence expectation than the baseline under LT model, and reduces 3.183 more under IC model.

Figure 6 shows the effect of the number of simulations n_{test} on the performance of the greedy algorithm. Intuitively, a larger n_{test} contributes to better performance but requires longer running time, which is consistent with the experiment result. Furthermore, we can see from the figure that about 4000 simulations are enough for this small-scale graph.



(a) Results of DER-LT



(b) Results of DER-IC

Figure 5. The influence expectation during the process of removing edges one by one, using the baseline max weight/probability algorithm or greedy Algorithm 1. The number of simulations $n_{test} = 5000$

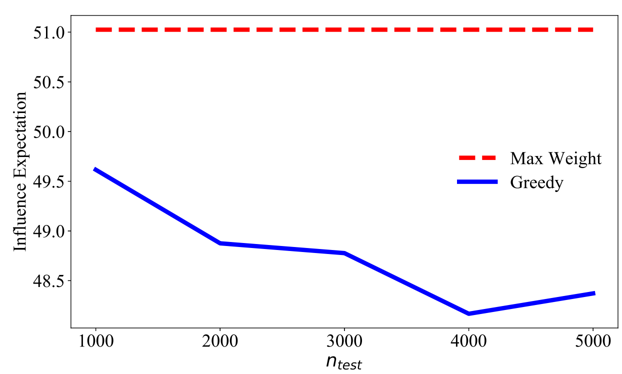


Figure 6. The influence expectation after removing l edges via max weight algorithm or greedy Algorithm 1, with different number of simultation times n_{test}. The diffusion model is LT, and the baseline algorithm does not need simulation, thus unrelated with n_{test} in the figure.

6.2. Experiments on CER

For experiments on CER, we set regularization weight $\lambda = 30$, step size $\eta = 0.1/\lambda$, the number of iterations $n_{it} = 15$, perturbation $\delta = 0.1$, and the number of simulations $n_{test} = 5000$ by default. We let step size η decrease when λ increase to prevent the gradient of the regularization term from varying violently when λ changes. This also follows the intuition that the real edge matrix $\hat{\mathbf{E}}$ will not deviate far from the predicted edge matrix $\mathbf{E}$ when λ is large. We can not think of any naive baseline algorithms for the CER problem, and thus present only the results of the proposed SGD algorithm.

Figure 7 shows the performance of the proposed SGD algorithm on CER-LT. From Figure 7(c), we can see that the SGD process on the objective f_C converges very fast, in only about 5 iterations. This validates that the proposed SGD algorithm can converge at least to a local minimum of the objective f_C. The influence when f_C is minimized is the robust influence under the CER formulation. Intuitively, larger λ implies more accurate estimation of the real social graph, and more robust performance of the computed IM strategy. Both Figures 7(b) and 7(c) validates this conjecture. With larger λ, the influence expectation when f_C is minimized, is larger. Figure 8 shows the performance of the SGD algorithm on CER-IC, which also validates the convergence of the algorithm.

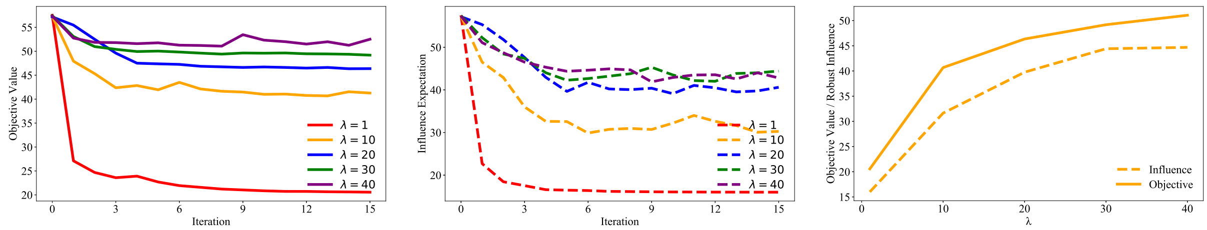
Figure 9 shows the effect of n_{test} on the performance of the SGD algorithm on CER-LT. The results validate the intuition that larger n_{test} contributes to more accurate numeracal gradient, and thus more stable SGD process. From the figure, we can also see that $n_{test} = 3000$ is enough for the small-scale network.

7. Future Work

This work sheds light on the robustness evaluation of IM strategies. Future work includes proving the NP-hardness under LTM, which we fail to manage. Since we study the robustness evaluation problem from scratch, the proposed greedy and SGD algorithms are with relatively high time complexity, and we expect to address this in the future. Furthermore, this work lay a solid fundation for designing robust IM algorithms against edge uncertainty, which may be studied in the future.

8. Conclusion

In this work, we investigate the problem of evaluating the robustness of IM strategies. We give a reasonable formulation of it, and specifically discuss the discrete and the continuous case. For the discrete case, we prove the NP-harness under IC and DLT diffusion models, and design a greedy algorithm with $1 - 1/e$ approximation ratio under LT model. Furthermore, empirical experiments show that the greedy algorithm outperforms the naive baseline, and validate the theoretical results. For the continuous case, we design a SGD algorithm, implement it, and validate its feasibility by applying to a simulated social network.



(a) Objective function f_C's value with different λ.

(b) Influence expectation with different λ

(c) The minimum f_C during the SGD process and the corresponding influence expectation with different λ

Figure 7. Experiments on CER-LT with different λ.

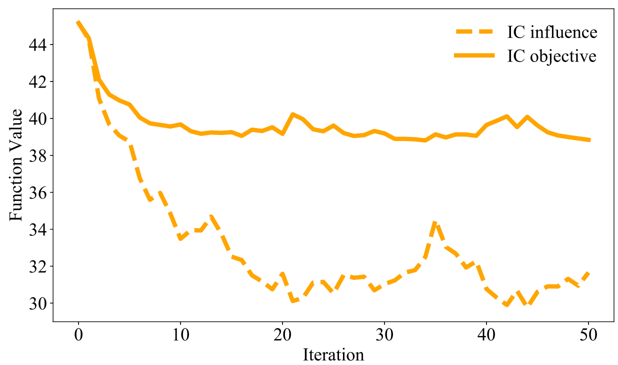


Figure 8. The objective function f_C's value and the influence expectation during the SGD process under IC model. Regularization weight $\lambda = 30$, step size $\eta = 0.0033$, number of simulations $n_{test} = 10000$.

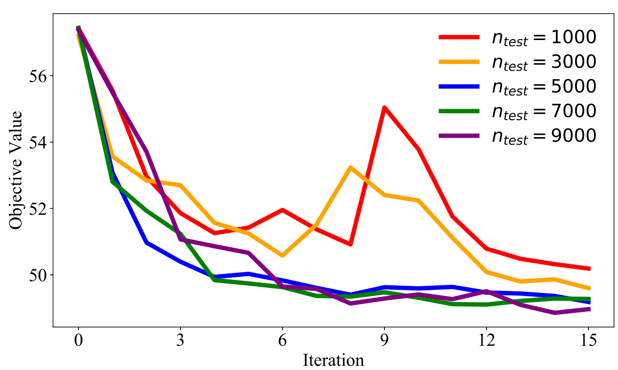


Figure 9. The objective function f_C's value during the SGD process under LT model, with different number of simulations n_{test}.

References

Bogunovic, I., Mitrović, S., Scarlett, J., and Cevher, V. Robust submodular maximization: A non-uniform partitioning approach. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 508–516. JMLR. org, 2017.

Chen, S., Yin, X., Cao, Q., Li, Q., and Long, H. Targeted influence maximization based on cloud computing over big data in social networks. *IEEE Access*, 8:45512–45522, 2020.

Chen, W., Lin, T., Tan, Z., Zhao, M., and Zhou, X. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 795–804, 2016.

Cohen, E., Delling, D., Pajor, T., and Werneck, R. F. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, pp. 629–638. ACM, 2014.

Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., and Xie, K. Simulated annealing based influence maximization in social networks. In *AAAI*. AAAI Press, 2011.

Jiang, Y. and Zhao, X. Adaptive influence maximization based on community. *EE447 project*, 2019.

Kalimeris, D., Kaplun, G., and Singer, Y. Robust influence maximization for hyperparametric models. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3192–3200. PMLR, 2019.

Kempe, D., Kleinberg, J. M., and Tardos, É. Maximizing the spread of influence through a social network. In *KDD*, pp. 137–146. ACM, 2003.

Li, M., Wang, X., Gao, K., and Zhang, S. A survey on information diffusion in online social networks: Models and methods. *Information*, 8(4):118, 2017.

McAuley, J. J. and Leskovec, J. Learning to discover social circles in ego networks. In *NIPS*, pp. 548–556, 2012.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Song, G., Zhou, X., Wang, Y., and Xie, K. Influence maximization on large-scale mobile social network: A divide-and-conquer method. *IEEE Trans. Parallel Distrib. Syst.*, 26(5):1379–1392, 2015.

Xavier, E. C. A note on a maximum k-subset intersection problem. *Inf. Process. Lett.*, 112(12):471–472, 2012.