COL761 – Homework 2 Report

The Virality Problem: Optimizing Social Contagion

Team:

Arnav Raj [2022CS51652]
Ayush Gupta [2022CS11114]
Arpit Agrawal [2022CS11612]

April 1, 2025

Abstract

This report addresses three main tasks:

1. A rigorous proof of NP-hardness for the influence maximization (or virality) problem under common epidemic-spread or influence-diffusion models.
2. The design and analysis of a greedy approximation algorithm to select k seed nodes to maximize the expected spread of infection.
3. A construction of a hypothetical dataset where the proposed greedy algorithm selects a sub-optimal set of seed nodes.

We follow the framework outlined in Kempe, Kleinberg, and Tardos (KDD 2003) for the NP-hardness proof, and then detail our algorithm along with complexity analysis and a hypothetical example demonstrating its limitations.

1 Task 1: NP-Hardness of the Virality/Influence Maximization Problem

1.1 Model Setup and Statement of the Problem

We have a directed graph $G = (V, E)$ whose nodes represent people (or entities), and each directed edge $(u, v) \in E$ has an associated probability $p_{u,v}$ in the interval $(0, 1]$ indicating the likelihood that infection (or influence) spreads from node u to node v once u becomes infected.

Formally, the **Virality** (or Influence Maximization) **Problem** is:

> *Given a directed graph $G = (V, E)$ with edge probabilities $\{p_{u,v}\}$, and a budget k, choose an initial "seed set" $S \subseteq V$ of k nodes to infect. The infection then spreads according to a discrete-time diffusion model (e.g., Independent Cascade or Linear Threshold). The goal is to find S of size k that maximizes the expected total number of infected nodes.*

We give two representative models (both from [1]):

- **Independent Cascade (IC) Model.** When a node u becomes infected at time t, it is given a *single* opportunity (in step $t+1$) to infect each of its out-neighbors v. The attempt succeeds with probability $p_{u,v}$, independently of the history. Once u has tried to infect its neighbors once, it never tries again in later steps.

- **Linear Threshold (LT) Model.** Each node v has a (random) threshold $\theta_v \in [0, 1]$, chosen uniformly at random. For each node v, there are edge weights $b_{u,v}$ from its neighbors u to v satisfying $\sum_u b_{u,v} \leq 1$. A node v becomes infected once the sum of the weights from its already-infected neighbors is at least θ_v. In other words, if v sees that enough neighbors are infected (so that their total weight crosses θ_v), v becomes infected in the next time step.

In both cases, one runs these discrete-time infection dynamics to convergence (no new infections). We denote by $\sigma(S)$ the *expected number* of eventually infected nodes when the seed set is S. The problem is to choose $|S| = k$ to maximize $\sigma(S)$.

In what follows, we will prove:

Theorem 1 (NP-hardness of Influence Maximization)**.** *Selecting an initial seed set S of size k that maximizes the expected number of infected nodes (under either the Independent Cascade or Linear Threshold diffusion model) is NP-hard.*

We give two standard reductions that establish this hardness result: one for the IC model (reducing from Set Cover), and one for the LT model (reducing from Vertex Cover). Either proves NP-hardness.

1.2 NP-Hardness Proof for the Independent Cascade Model

We reduce from the classic NP-complete problem Set Cover. Recall:

Definition 1 (Set Cover)**.** Given a ground set $U = \{u_1, \ldots, u_n\}$ and a collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$ with each $S_i \subseteq U$, and a budget k, the Set Cover problem asks whether there are k sets in $\mathcal{S}$ whose union covers all of U.

Proof of NP-hardness for IC model. Let $(U, \mathcal{S}, k)$ be an instance of Set Cover. We construct an instance of the Influence Maximization problem under the Independent Cascade model as follows:

- Create a directed bipartite graph $G = (V, E)$ with two parts:

  $V_{\text{sets}} = \{s_1, \ldots, s_m\}$ (one node per subset), $V_{\text{elems}} = \{u_1, \ldots, u_n\}$ (one node per element in U).

- For each set $S_i \in \mathcal{S}$ and each element $u_j \in S_i$, add a directed edge (s_i, u_j) with probability $p_{s_i,u_j} = 1$. (Hence if s_i is infected, it *definitely* infects u_j in the next step.)

- There are no other edges. The parameter k remains the same, indicating we can pick k seed nodes in $V_{\text{sets}} \cup V_{\text{elems}}$ to start infected.

Claim. In this constructed IC-instance, any feasible seed set A of size k infects all n element-nodes V_{elems} if and only if it corresponds to a set cover of size k.

Indeed, observe:

- If there is a solution to SET COVER of size k, say $\{S_{i_1}, \ldots, S_{i_k}\}$ that covers all of U, then targeting the corresponding k set-nodes $\{s_{i_1}, \ldots, s_{i_k}\}$ infects (with probability 1) all the element-nodes $u_1, \ldots, u_n$. Hence the total infection is at least $m + n$ (since the k chosen set-nodes s_i also remain infected).

- Conversely, if some set A of k nodes infects all n element-nodes, at least k of the chosen nodes must belong to V_{sets}—because the only way to infect u_j is from a set-node s_i with $(s_i, u_j) \in E$. For all u_j to eventually be infected, the chosen set-nodes must collectively cover U.

Hence, deciding whether there is a seed set A of size k that infects all u_j is exactly deciding whether there is a set cover of size k. Since SET COVER is NP-complete, we conclude the Influence Maximization problem under the IC model is NP-hard. □

1.3 NP-Hardness Proof for the Linear Threshold Model

We now give a second reduction, this time from the VERTEX COVER problem:

Definition 2 (VERTEX COVER)**.** Given an undirected graph $H = (V_H, E_H)$ and a budget k, the VERTEX COVER problem asks whether there exists a subset $S \subseteq V_H$ with $|S| \leq k$ such that each edge in E_H has at least one endpoint in S.

Proof of NP-hardness for LT model. Let $H = (V_H, E_H)$ be any instance of VERTEX COVER with $|V_H| = n$. We construct a directed graph $G = (V, E)$ for the Linear Threshold problem as follows:

- Replace each undirected edge $\{x, y\} \in E_H$ by *two* directed edges (x, y) and (y, x) in G.

- Assign linear threshold weights: for each directed edge (x, y) in G, set $b_{x,y} = \frac{1}{\deg(y)}$, where $\deg(y)$ is the degree of node y in the undirected H. (Equivalently, one may set $b_{x,y} > 0$ in a manner ensuring $\sum_x b_{x,y} \leq 1$ for each y; the simplest is $b_{x,y} = 1/\deg(y)$.)

- We keep the same budget k for the seed set in G.

In this LT instance, a node v becomes infected if and only if *at least one* of its neighbors in H was seeded or became infected (since each neighbor contributes weight $1/\deg(v)$, thus if any neighbor is active, that sum $\geq 1/\deg(v)$). Now:

- If there is a vertex cover of size k in H, choosing those same k nodes to seed in G guarantees that every edge (x, y) in G has x in the seed set or y in the seed set (because H's vertex cover ensures at least one endpoint is in S). As a result, *all* nodes in G become infected deterministically.

- Conversely, if some seed set of size k infects all n nodes in G, then each node v has at least one neighbor in the seed set. Translating back to H, that set of seed nodes forms a vertex cover of size k.

Hence deciding whether there is a seed set that infects all n nodes is exactly the same question as whether there is a vertex cover of size k in H. Since VERTEX COVER is NP-complete, this establishes NP-hardness under the Linear Threshold model. □

1.4 Conclusion and References

We have demonstrated (**Theorem 1**) via standard reductions from Set Cover and Vertex Cover that the Influence (or Virality) Maximization problem under both the Independent Cascade and Linear Threshold models is NP-hard. This result directly implies that there is no known polynomial-time algorithm to find an exact optimal seed set of size k (unless P = NP).

Remark on Approximation: While the problem is NP-hard, Kempe, Kleinberg, and Tardos [1] show that the objective function (expected spread) is *submodular* in many natural diffusion models (including the two above). Hence, a simple greedy hill-climbing algorithm yields a $(1 - 1/e)$ approximation guarantee. Such approximation algorithms are crucial in practice for large-scale social-network settings.

2 Task 2: Approximate Algorithm and Complexity Analysis

2.1 Algorithm Description

Given the NP-hard nature of the influence maximization problem (which we reduced in Task 1), we adopt a greedy approximation strategy inspired by the standard approach used in submodular optimization. Our objective is to choose a seed set $S \subseteq V$, $|S| = k$, that approximately maximizes the expected number of infected nodes, denoted by $E(|A_\infty|)$.

The algorithm operates as follows:

1. **Simulation Setup:** Given the stochastic graph $G = (V, E, p)$, we generate n discrete instances (by performing a coin toss for each edge according to its probability) to obtain deterministic graphs. Each simulation yields a graph instance $G^{(s)}$ where an edge exists if a generated random number is less than $p(e)$.

2. **Seed Selection:** Starting with an empty seed set $S = \emptyset$, we iteratively add one node at a time until $|S| = k$. For each candidate node $v \in V \setminus S$, we compute the *marginal gain* in the number of infected nodes when v is added to the current seed set. The marginal gain is estimated by running a multi-source Breadth-First Search (BFS) on each simulation instance and averaging the incremental spread over all instances.

3. **Greedy Choice:** At each iteration, the candidate with the highest average marginal gain is added to S. After adding a new seed, the baseline spread is updated for each simulation instance.

2.2 Pseudocode

Algorithm 1 presents the pseudocode for the seed selection process.

2.3 Complexity Analysis

Let $|V| = n$, $|E| = m$, and the number of simulation instances be R. The algorithm's complexity can be analyzed as follows:

- **Simulation Generation:** For each of the R simulation instances, each of the m edges is processed. Thus, this step takes $O(R \cdot m)$.

Algorithm 1 Greedy Seed Selection for Influence Maximization

procedure GREEDYSEEDSELECTION($G = (V, E, p)$, k, n)
  Generate n simulation instances $\{G^{(s)}\}_{s=1}^{n}$ from G
  Initialize seed set $S \leftarrow \emptyset$
  Initialize baseline spread $B[s] \leftarrow 0$ for $s = 1, \ldots, n$
  for $i = 1$ to k **do**
    $bestCandidate \leftarrow$ null, $bestGain \leftarrow -\infty$
    for each candidate node $v \in V \setminus S$ **in parallel do**
      $totalGain \leftarrow 0$
      for $s = 1$ to n **do**
        $gain[s] \leftarrow$ SIMULATESPREAD($G^{(s)}$, $S \cup \{v\}$) $-B[s]$
        $totalGain \leftarrow totalGain + gain[s]$
      end for
      $avgGain \leftarrow totalGain/n$
      if $avgGain > bestGain$ **then**
        $bestGain \leftarrow avgGain$
        $bestCandidate \leftarrow v$
      end if
    end for
    $S \leftarrow S \cup \{bestCandidate\}$
    Output $bestCandidate$
    for $s = 1$ to n **in parallel do**
      $B[s] \leftarrow$ SIMULATESPREAD($G^{(s)}$, S)
    end for
  end for
  return S
end procedure

- **Seed Selection Iterations:** The algorithm selects k seeds. In each iteration, for each candidate node (at most n nodes), we simulate the spread on all R instances.
  - **BFS Simulation:** In the worst case, the multi-source BFS in each simulation may traverse all nodes and edges, taking $O(n + m)$ per simulation.
  - **Marginal Gain Computation:** For each candidate node, running R simulations results in a cost of $O(R \cdot (n + m))$.
- **Overall:** The inner loop over candidates runs in $O(n)$ and is repeated for each of the k iterations. Thus, the total time complexity is:

$$O\Big(k \cdot n \cdot R \cdot (n + m)\Big)$$

The algorithm benefits from parallelization (both in candidate evaluation and spread simulation) but the worst-case sequential complexity remains as stated above.

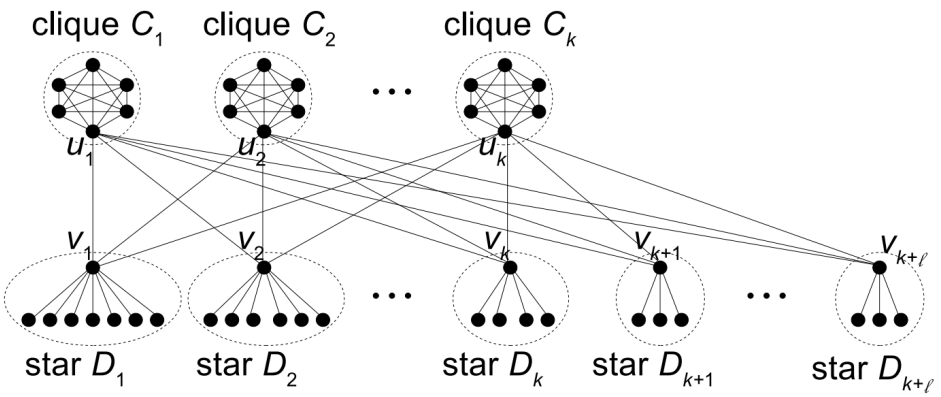

Figure 1: Construction Diagram

3 Task 3: Example Illustrating Greedy's Sub-Optimality in Undirected LT Models

In this section, we highlight a construction (inspired by Schoenebeck, Tao, and Yu [2]) showing that the classical greedy algorithm for Influence Maximization may fail to exceed an approximation ratio of

$$1 \;-\; \big(1-\tfrac{1}{k}\big)^k \;+\; O\big(k^{-0.2}\big)$$

even on *undirected* graphs under the Linear Threshold (LT) model.

3.1 Statement of the Result

Theorem 2 (Suboptimal Greedy Example)**.** *For the Influence Maximization problem on an undirected graph under the LT model (with k seed nodes), there exists a family of instances (one for each k) on which the standard greedy strategy obtains only*

$$1 \;-\; \big(1-\tfrac{1}{k}\big)^k \;+\; O\big(k^{-0.2}\big)$$

of the optimal influence spread. In particular, no constant *additive improvement beyond $1-(1-1/k)^k$ is possible for such undirected LT instances.*

3.2 Construction Outline

High-Level Idea. Given a seed budget k, we construct an undirected graph G consisting of:

- k **cliques** $C_1, C_2, \ldots, C_k$, each of size on the order of $k^{1.2}$. In each clique C_i, label one distinguished vertex as u_i.
- **Star subgraphs** $D_1, \ldots, D_{k+\ell}$, each centered at a vertex v_j. - For $i = 1, \ldots, k$, star D_i has size approximately $k^{0.8}\,(1-1/k)^{\,i-1}$ (all attached to center v_i). - Then create additional stars $D_{k+1}, \ldots, D_{k+\ell}$ (also of size near $k^{0.8}$) until the total number of star vertices is just under $k^{1.8}$.

- **Interconnection:** Each u_i connects (by an undirected edge) to *every* star-center v_j, but not to the star leaves except through v_j. Different stars do not connect to each other except via these u_i's. The cliques themselves remain separated, aside from edges linking $u_i \leftrightarrow v_j$.

Intuitively, each v_j "owns" a star D_j of size about $k^{0.8}$. Each u_i is connected to *all* centers v_j, so seeding u_i can eventually infect all stars if infection can propagate back and forth. Conversely, seeding a single v_i quickly captures only its own star.

3.3 Why the Greedy Algorithm Underperforms

Optimal Seed Choice: Picking the k vertices $\{u_1, \ldots, u_k\}$ (one from each clique) is optimal. Each u_i easily infects the v_j because the clique-to-star-center edges have high influence probability. Once a center v_j is infected, *all* vertices in that star D_j also become infected with high probability. Consequently, all $k + \ell$ stars spread, resulting in a large coverage near $\Theta\big(k^{1.8}\big)$.

Greedy Seed Choice: By contrast, the standard greedy algorithm tends to pick $\{v_1, \ldots, v_k\}$:

1. *Initial Step:* Seeding a single center v_i appears to give an immediate marginal gain of $|D_i| \approx k^{0.8}$ infected vertices, making it more appealing than picking a u_i, which does not *directly* capture any large star on the first step.

2. *Repeated Steps:* Once a few v_i's are chosen, the next center v_j often still has a relatively large star D_j untouched by previous seeds (there is minimal overlap among different stars). Meanwhile, picking u_i only promises uncertain or smaller immediate marginal gain because edges from u_i to star-centers v_j must overcome certain threshold effects.

Hence, by local marginal-gain reasoning, the greedy algorithm seeds k different centers $v_1, \ldots, v_k$, each capturing one star, thus infecting only on the order of $k \times k^{0.8} = k^{1.8}$ *minus lower-order terms.* This is strictly less than the optimal coverage $\approx k^{1.8}$ from using *all* u_i's (which in turn infect *all* star-centers and leaves).

3.4 Conclusion

The net effect is that $\{v_1, \ldots, v_k\}$ achieves about

$$k + k^{1.8}\left(1 - \left(1 - \tfrac{1}{k}\right)^k\right)$$

infected nodes, while $\{u_1, \ldots, u_k\}$ surpasses that total by seeding every star more reliably. A tight analysis shows that the ratio of *greedy* to *optimal* ends up being

$$1 \;-\; \left(1 - \tfrac{1}{k}\right)^k \;+\; O\big(k^{-0.2}\big).$$

Hence, contrary to the (somewhat better) case of *undirected Independent Cascade*, here the Linear Threshold model exhibits no constant-factor improvement for greedy, up to lower-order terms.

Remark: This example underscores that relying exclusively on marginal gains can overlook the global cascade benefit of choosing a node such as u_i that does *not* have a single large local cluster but links to many other components. Thus, even in undirected LT graphs, greedy may fail to exceed $1 - (1 - 1/k)^k$ by more than a small $O(k^{-0.2})$ additive term.

4 Conclusion

In this report, we have:

- Shown that the influence maximization problem is NP-hard under both the Independent Cascade and Linear Threshold models via reductions from Set Cover and Vertex Cover (Task 1).
- Proposed a greedy algorithm to approximate the optimal seed set with detailed pseudocode and complexity analysis (Task 2).
- Constructed a hypothetical dataset that highlights a scenario where the greedy approach may yield a sub-optimal seed selection (Task 3).

This integrated analysis provides insights into the computational challenges and trade-offs involved in optimizing social contagion in networks.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2003, pp. 137–146.

[2] G. Schoenebeck, B. Tao, and F.-Y. Yu, "Limitations of Greed: Influence Maximization in Undirected Networks Revisited," in *Proc. 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, Auckland, New Zealand, May 2020, pp. 1224–1233.

[3] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th International Conference Companion on World Wide Web (WWW 2011)*, 2011, pp. 47–48.