COL761 Assignment 2

Sanchit Singla - 2021ME21063, Shourya Dixant - 2023MCS2481

1 Task 1 - Problem Statement

The problem is stated as follows. A directed graph

$$G = (V, E, p),$$

is given, where each edge $e \in E$ is associated with a probability $p(e) \in (0, 1]$. These probabilities indicate the chance that the disease (or information) is transmitted from one node to another. Additionally, a budget of k vaccines is available. The task is to choose a seed set $A_0 \subset V$ with $|A_0| = k$ such that the expected number of infected nodes at the end of the diffusion process, denoted by $E(A_\infty)$, is maximized.

The diffusion process works as follows:

- If a node u gets infected at time t, it has exactly one opportunity at time $t + 1$ to infect each of its uninfected neighbors v with probability $p(u, v)$.
- Once u fails to infect a neighbor v, no further attempts are made.
- If u successfully infects v, then v becomes infected at time $t + 1$.

1.1 Reduction from Maximum Coverage

To establish NP-hardness, a reduction from the well-known maximum coverage problem to the influence maximization problem is presented. Since the maximum coverage problem is NP-hard, a polynomial-time reduction demonstrates that the influence maximization problem is NP-hard as well.

1.2 The Maximum Coverage Problem

The maximum coverage problem is defined as follows:

- **Input:** A finite universe $U = \{e_1, e_2, \ldots, e_m\}$ and a collection of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ with each $S_i \subseteq U$, along with an integer k.
- **Objective:** Select a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq k$ such that the number of covered elements,

$$\left| \bigcup_{S_i \in \mathcal{S}'} S_i \right|,$$

is maximized.

1.3 Graph Construction

Given an instance of maximum coverage, a directed bipartite graph $G = (V, E)$ is constructed as follows:

1. **Vertex Sets:**
   - For every set $S_i \in \mathcal{S}$, create a vertex a_i. This collection of vertices forms the left vertex set A.
   - For every element $e_j \in U$, create a vertex b_j. This forms the right vertex set B.
2. **Edges and Probabilities:** For every set S_i and for every element $e_j \in S_i$, add a directed edge from a_i to b_j with activation probability $p(a_i, b_j) = 1$. No other edges are added.
3. **Seed Selection:** The seed set A_0 is restricted to the vertices in A only, with the budget constraint $|A_0| = k$.

The construction runs in polynomial time because:

- The total number of vertices is $|V| = |A| + |B| = n + m$.
- The number of edges is at most $\sum_{i=1}^{n} |S_i|$, which is polynomial in the size of the input.

1.4 Equivalence of Objectives

The objective of influence maximization in the constructed graph is equivalent to that of the maximum coverage problem. Under the independent cascade model:

1. Initially, the seed set $A_0 \subseteq A$ (with $|A_0| = k$) is infected.
2. At time $t + 1$, every vertex $b_j \in B$ that is adjacent to some vertex $a_i \in A_0$ becomes infected with certainty because $p(a_i, b_j) = 1$.

Thus, the final set of infected nodes is:

$$A_0 \cup \{b_j \in B \mid \exists\, a_i \in A_0 \text{ such that } (a_i, b_j) \in E\}.$$

Define the function $\sigma(A_0)$ as the total number of infected nodes:

$$\sigma(A_0) = |A_0| + \left| \bigcup_{a_i \in A_0} \{b_j : (a_i, b_j) \in E\} \right|.$$

Since each vertex a_i represents a set S_i and each b_j represents an element e_j, this can be rewritten as:

$$\sigma(A_0) = k + \left| \bigcup_{a_i \in A_0} S_i \right|.$$

Because k is fixed, maximizing $\sigma(A_0)$ is equivalent to maximizing

$$\left| \bigcup_{a_i \in A_0} S_i \right|,$$

which is exactly the objective of the maximum coverage problem.

Conclusion

A polynomial-time reduction from the maximum coverage problem to the influence maximization problem has been presented. Since an optimal solution to the influence maximization problem in the constructed graph (where every edge has probability 1) directly yields an optimal solution to the maximum coverage problem, and given that the maximum coverage problem is NP-hard, it follows that the influence maximization problem is NP-hard as well.

This reduction confirms that finding the optimal set of k nodes to maximize the expected spread of infection in a network is NP-hard.

2 Task 2: Approximation Algorithm and Its Complexity

Since the influence maximization objective is both monotone and submodular (a property that can be shown via the analysis in Task 1), a simple yet effective approximation algorithm is the *greedy algorithm*. This algorithm is inspired by the greedy approach for the maximum coverage problem, which is known to achieve a $1 - \frac{1}{e}$ approximation guarantee.

2.1 Greedy Algorithm Description

The greedy algorithm works as follows:

1. Initialize the seed set $A_0 = \emptyset$.
2. For $i = 1$ to k:
   (a) For every node $v \in V \setminus A_0$, compute the *marginal gain*:
   $$\Delta(v \mid A_0) = E\big(A_0 \cup \{v\}\big) - E(A_0),$$
   where $E(X)$ is the expected number of infected nodes when X is the seed set.
   (b) Select the node v^* that maximizes the marginal gain, i.e.,
   $$v^* = \arg\max_{v \in V \setminus A_0} \Delta(v \mid A_0).$$
   (c) Update the seed set: $A_0 \leftarrow A_0 \cup \{v^*\}$.
3. Return A_0 as the final seed set.

Because the influence function $E(\cdot)$ is monotone (adding nodes does not decrease the spread) and submodular (the marginal gain decreases as the seed set grows), the greedy algorithm is guaranteed to achieve an approximation factor of at least $1 - \frac{1}{e}$ compared to the optimal solution [**?**].

2.2 Complexity Analysis

The overall complexity of the greedy algorithm depends on the following factors:

- There are k iterations, one for each node added to the seed set.

- In each iteration, the marginal gain is computed for each candidate node $v \in V \setminus A_0$. In the worst-case, this involves up to $|V|$ computations.
- The computation of the marginal gain $\Delta(v \mid A_0)$ does not have a closed form in general and is often estimated using Monte Carlo simulations. Let T be the number of simulations used to estimate the expected spread.

Thus, the time complexity can be expressed as:

$$O\big(k \cdot |V| \cdot T\big).$$

2.3 Summary

An efficient approximate algorithm for the influence maximization problem is the greedy algorithm that iteratively selects the node with the largest marginal gain in the expected spread. Given the monotonicity and submodularity of the expected spread function, the greedy algorithm achieves a $\big(1 - \frac{1}{e}\big)$ approximation guarantee, with a time complexity of $O(k \cdot |V| \cdot T)$ under a Monte Carlo simulation framework.

3 Task 3: Hypothetical Dataset Illustrating Suboptimal Greedy Selection

Consider a network with three candidate seed nodes: A, B, and C, and suppose the task is to select $k = 2$ nodes. The influence (or coverage) of each node is defined by the set of nodes it can activate. In this example, the influence sets are as follows:

- **Node A:** Covers a set X with $|X| = 60$ nodes.
- **Node B:** Covers a set Y with $|Y| = 60$ nodes, with an overlap with X of $|X \cap Y| = 10$ nodes.
- **Node C:** Covers a set Z with $|Z| = 80$ nodes. However, the overlap with the sets of both A and B is high; specifically, assume

$$|Z \cap X| = |Z \cap Y| = 55.$$

3.1 Greedy Algorithm Behavior

The greedy algorithm selects nodes based on their individual spread and marginal gain:

1. **Iteration 1:**
   - The individual influence of each node (including the node itself) is approximately:

$$\begin{aligned} \sigma(A) &\approx 60 + 1 = 61, \\ \sigma(B) &\approx 60 + 1 = 61, \\ \sigma(C) &\approx 80 + 1 = 81. \end{aligned}$$

   - Since node C has the highest individual spread, it is selected first.

2. **Iteration 2:**

   - Now, the marginal gain of adding either A or B is calculated with respect to the current seed set $\{C\}$. For node A, the new nodes it would cover are those in $X \setminus Z$. Given that $|X| = 60$ and $|X \cap Z| = 55$, the additional unique nodes contributed by A are:

   $$|X \setminus Z| = 60 - 55 = 5.$$

   - A similar calculation applies for node B. Thus, the marginal gain of adding either A or B is only about 6 nodes.
   - Consequently, the final seed set chosen by the greedy algorithm is $\{C, A\}$ (or $\{C, B\}$), yielding an approximate total spread of:

   $$\sigma(\{C, A\}) \approx 81 + 6 = 87.$$

3.2 Optimal Selection

In contrast, consider the alternative selection of nodes A and B. Their union covers:

$$|X \cup Y| = |X| + |Y| - |X \cap Y| = 60 + 60 - 10 = 110.$$

Including the seeds themselves (if not already counted), the overall spread is significantly higher than the spread achieved by the greedy algorithm.

3.3 Conclusion

This hypothetical dataset demonstrates that the greedy algorithm, by selecting node C first due to its high individual spread, ends up with a seed set $\{C, A\}$ that covers roughly 87 nodes. However, the optimal pair $\{A, B\}$ would yield a union covering 110 nodes. This example illustrates a scenario where the greedy algorithm selects suboptimal nodes because the high overlap between node C and the other candidates drastically reduces the marginal gains in the second iteration.