Information Exposure Maximization Project

Zhiyuan Zhong
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
12110517@mail.sustech.edu.cn

I. Introduction

This project addresses Information Exposure Maximization (IEM), a key problem in social influence analysis within social networks. IEM involves selecting two user sets (campaigns) to maximize the expected number of users reached by both campaigns or unaffected by either.

II. Preliminary

Information Exposure Maximization (IEM) is formulated as an optimization problem within the context of social influence analysis on a given social network. Let $G = (V, E)$ represent the social network, where V is the set of vertices (users) and E is the set of edges representing social connections. The goal of the IEM problem is to find two sets of users with the maximum balanced information exposure. In the following, several preliminary definitions are provided first, and then a formal definition and an example of the IEM problem are presented.

A. Definitions

- Social Network: $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ represents the node set, and $E = V \times V$ represents the edges between nodes.
- Campaigns: $C = \{c_1, c_2\}$ represents two campaigns, each holds a viewpoint.
- Initial Seed Set: $I_i \subseteq V$, $i \in \{1, 2\}$ represents the initial seed set for campaigns c_i.
- Balanced Seed Set: $S_i \subseteq V$, $i \in \{1, 2\}$ represents the target seed set that you need to find for each campaign c_i.
- Budget: k represents the size of the target seed set; $|S_1| + |S_2| \leq k$.
- Diffusion Probability: $P = \{P^i_{(u,v)} | (u, v) \in E_i$, $i \in \{1, 2\}$ represents the edge weight associated with campaign c_i, where p_i represents the probability of node u activating node v under each campaign c_i.
- Diffusion Model: M captures the stochastic process for seed set $U_i = I_i \cup S_i$ spreading information on G. We assume that information on the two campaigns propagates in the network following the independent cascade (IC) model. The two campaigns' messages propagate independently of each other (such propagation is often called heterogeneous propagation). The diffusion process of the first campaign (the process for the second campaign is analogous) unfolds in the following discrete steps:
  i. In step $t = 0$, the nodes in seed set U are activated, while the other nodes stay inactive;
  ii. Each active user u for campaign c_i in step t will activate each of its outgoing neighbor v that is inactive for campaign c_i in step $t-1$ with probability p_i; The activation process can be considered as flipping a coin with head probability p_i: if the result is heads, then v is activated; otherwise, v stays inactive; Note that u has only one chance to activate its outgoing neighbors for campaign c_i. After that, u stays active and stops the activation for campaign c_i;
  iii. The diffusion instance terminates when no more nodes can be activated.
- Exposed Nodes: Given a seed set U, $r_i(U)$ is the set of vertices that are reached from U using the aforementioned cascade process for campaign c_i. Note that in one propagation, in addition to nodes that were successfully activated by U, nodes that were once attempted to be activated but were not successfully activated by U are also considered to be reached by U. Since the diffusion process is random, $r_i(U)$ is a random variable.

B. Problem Formation

Given a social network $G = (V, E)$, two sets I_1 and I_2 of initial seeds for the two campaigns, and a budget k, the Information Exposure Maximization (IEM) problem is to find two sets S_1 and S_2, where $|S_1| + |S_2| \leq k$, and maximize the expected number of vertices that are either reached by both campaigns or remain oblivious to both campaigns, i.e.,

$$\max \Phi(S_1, S_2) = \max \mathbb{E}[|V \setminus r_1(I_1 \cup S_1) \triangle r_2(I_2 \cup S_2))|]$$

$$s.t. |S_1| + |S_2| \leq k,$$

$$S_1, S_2 \subseteq V,$$

where $\triangle$ denotes the symmetric difference, $A \triangle B = (A \setminus B) \cup (B \setminus A)$.

III. Methodology

A. Heuristic Search

The heuristic search aims to iteratively select nodes to maximize information exposure in a social network. It involves initializing seed sets for two campaigns, expanding these sets while considering network influence, and optimizing the selections based on their impact on information exposure. The algorithm iteratively adds nodes to the seed sets, choosing the node that maximizes the expected exposure when added to the current set. The exposure is estimated through Independent Cascade (IC) simulations.

Algorithm 1 Heuristic Search

$pre\_exposed1 \leftarrow$ Calculate exposed sets for each single node in campaign 1
$pre\_exposed2 \leftarrow$ Calculate exposed sets for each single node in campaign 2
$s1 \leftarrow \{\}$
$s2 \leftarrow \{\}$
while $|s1| + |s2| < k$ **do**
  $u1 \leftarrow i1 \cup s1$
  $u2 \leftarrow i2 \cup s2$
  $exposed\_1 \leftarrow$ Get exposed set for $u1$ in campaign 1
  $exposed\_2 \leftarrow$ Get exposed set for $u2$ in campaign 2
  $max1, max2, candidate1, candidate2 \leftarrow 0, 0, 0, 0$
  for $node$ in $nodes\_not\_in1$ **do** {Nodes not in $i1$ or $s1$}
    $temp\_exposed \leftarrow exposed\_1 \cup pre\_exposed1[node]$
    $val \leftarrow$ Get $\Phi(temp\_exposed,\ exposed\_2)$
    if $val > max1$ **then**
      $max1 \leftarrow val$
      $candidate1 \leftarrow node$
    end if
  end for
  for $node$ in $nodes\_not\_in2$ **do**
    $temp\_exposed \leftarrow exposed\_2 \cup pre\_exposed2[node]$
    $val \leftarrow$ Get $\Phi(exposed\_1,\ temp\_exposed)$
    if $val > max2$ **then**
      $max2 \leftarrow val$
      $candidate2 \leftarrow node$
    end if
  end for
  if $max1 > max2$ **then**
    $s1 \leftarrow s1 \cup \{candidate1\}$
  else
    $s2 \leftarrow s2 \cup \{candidate2\}$
  end if
end while

By using set operations and pre-processing possible propagation set, the heuristic search performs fast with high quality.

B. Evolutionary Algorithm

The evolutionary algorithm employs a combination of crossover, mutation, and fitness evaluation to evolve a population of potential solutions over multiple generations. It first simulate IC process to estimate the influence set of nodes of each single node and calculate the reward ϕ. Then it randomly selects $[0, k]$ nodes from the top $n/2$ nodes with the highest ϕ for $|population|$ times as the initial populations. Then it adopts single point crossover/uniform crossover and flip bit mutations to generate new offsprings. Finally it returns the best individual as the solution.

Algorithm 2 Evolutionary Algorithm

$population \leftarrow$ InitializeRandomPopulation(n, pop_size)
$best \leftarrow$ RandomChoice($population$)
for generation $\leftarrow$ 1 to $evol\_num$ **do**
  $offspring \leftarrow$ GenerateOffspring($population$)
  $candidates \leftarrow$ EvaluateFitness($population \cup offspring$)
  $candidates \leftarrow$ SortByFitness($candidates$)
  if $candidates[0]$.fitness $>$ get_fitness($best$) **then**
    $best \leftarrow candidates[0]$.solution
  end if
  $population \leftarrow$ SelectTopSolutions($candidates, pop\_size$)
  Print "Round generation :", $candidates[0]$.fitness
end for
$(s1, s2) \leftarrow$ ExtractBestSolution($best$)
Return $s1, s2$

C. Analysis

For the heuristic search method. It adopts a greedy principle to select the node with the best reward each round, which does not guarantee a optimal solution of the problem. The deciding factor of its performance is the estimation of exposure set of one node. Pre-process the exposure set can reduce computations in Monte-Carlo simulations and accelerate the algorithm. Calculating intersection/union/symmetric difference using sets are also much faster.

For the Evolutionary algorithm, the effectiveness of the crossover and mutation mechanisms significantly influences the algorithm's performance. Like population size and number of generations. For Fitness Evaluation, the accuracy of the fitness evaluation function determines the algorithm's ability to distinguish between good and poor solutions, where the same IC process is used. Compared with heuristic search, evolutionary algorithm is more random and unstable.

IV. Experiments

A. Setup

The datasets are provided in the lab. Run on python 3.10.

B. Results and Analysis

The result is on the OJ.

The heuristic search finds a better solution with faster speed than the evolutionary algorithm, since it requires more domain knowledge optimizations. Run IC process for 3 times seems to be a strategy that is time-efficient without sacrificing quality. For EA, finding a good initial solution (population) is critical, since the randomness of crossover and mutation could be unexpected.

V. Conclusion

The Heuristic Search Method, guided by a greedy principle, quickly selects nodes with the best rewards at each iteration. While providing fast solutions, its lack of optimality and sensitivity to initial conditions must be noted. Efficient preprocessing of exposure sets, leveraging set operations, significantly accelerates Monte-Carlo simulations.

On the other hand, the Evolutionary Algorithm, offering flexibility and adaptability, balances exploration and exploitation through crossover and mutation. While not guaranteeing global optimality, it handles complex solution spaces. Fine-tuning parameters is crucial for its performance, which is very hard and tricky.