Artificial Intelligence Project 2: Influence Maximization Problem

Cai Yufei 11812101
Computer Science and Technology
Southern University of Science and Technology
caiyiwen@caiyiwen.tech

1. Preliminaries

1.1. Problem Description

Influence maximization is to find a group of nodes under a specific network propagation model to maximize the final influence of this group of nodes.

Given a directed graph G like a social network, the influence maximization problem is define as to find k nodes in G whose adoptions of a certain idea or trigger the largest expected number of follow-up adoptions by the remaining nodes. The problem is first raised up in paper [1], and it has been proved to be a NP complete problem. There are two kind of stochastic diffusion models. They are LT(Linear Threshold) and IC(Independent Cascade).

In this project, we are going to study and research the problem by two main aspects. The first aspect is to use two different models which are popular to evaluation the answers of the problem. The second aspect is to study and design efficient algorithms to solve the problem as effective and efficient as possible and test the results of the algorithm(s).

1.2. Problem Applications

In modern society, information is the most important. If we currently have a piece of information and we want to maximize the spreading scale of information, we have to consider how to choose the entry point and choose who is the node of initial dissemination so that the entire network can be most affected by the information.

Let us imagine such a background - a company wants to promote a new product to the market, hoping that it can be accepted by most people in the network. The company plans to initially target a small group of people, and then send them free product samples (the products are very expensive, so the company has to limit the budget and only select a small group of people to distribute). The company hopes that these initially selected users can recommend these products to their friends, and their friends will influence their friends' friends. If this continues, many individuals will eventually be affected to accept these new products. The powerful influence of the world through these word of mouth can be called a virus market. The problem here is how to choose a collection of independent individuals to distribute free items so that they can influence the largest number of people in social networks.

Influence Maximization Problem is a basic and typical problem in big data and statistics. Under so many circumstances, the goal of businessmen or journalists is to make their merchandises or news spread around on a specific group whose scales are as large and wide as possible. To achieve this goal, people have to make use of the models and algorithms in this abstract problem. As a consequence, solving the problem efficiently and effectively will benefit for the economy and the control of public opinions when there are large quantities of requirements.

2. Methodology

2.1. Notation

Notation	Meaning
V	The set of all nodes in a graph
E	The set of all edges in a graph
$G = (V, E)$	A graph whose node set is V and edge set is E
$n = \|V\|$	The number of nodes in graph which equals $\|V\|$
$m = \|E\|$	The number of edges in graph which equals $\|E\|$
$w(e)$	The weight of edge e
$d_{in}(u)$	The in-degree of node u
k	The number of seeds in IMP
S	The set of seeds in IMP
RR_i	The i-th RR set in IMM
R	The set of RR sets in IMM
ϵ	The ratio parameter of error in IMM
$C(S)$	The number of RR sets which intersect S in IMM
M	The influence model
$I(S, G, M)$	The influence measured by model M on graph G

2.2. Data Structure

Data Structure	Application
$\{node_i : list(), ...\}$	The adjacent list of a graph
$node_i : int$	Index of the i-th node
$node_i : list()$	The edges which start from $node_i$
$activity\_set : list$	List of nodes which are activated
$new\_activity\_set : list$	Iterative set of activated nodes
$seed\_set : list$	The seeds selected
max_heap	The maximum heap in IMM
$active\_set$	Sets which are active

2.3. Detail of Algorithm

First and foremost, we are supposed to solve the ISE(Influence Spread Evaluation) problem. As a result, we are required to design algorithms to simulate the process where influence spread with specific models which can be called stochastic diffusion model.

There are currently two models which are quite popular and widely used. One of them is IC(Independent Cascade) model, and the other is LT(Linear Threshold) model.

IC model considers the spread process as a random process on each edge which is unrelated to the nodes themselves. The nodes only serves as influence media which cannot determine where the influence will spread. Its process is Algorithm 1.

- When a node u gets activated, it possesses a single chance to activate each inactivate node v when $e = (u, v) \in E$. The probability of the activation is proportional to the weight $w((u, v))$.
- After then, the activated nodes remain its active state but do not contribute to the following activation any more.
- The weight of edge (u, v) is calculated as $w((u, v)) = \frac{1}{d_{in}(v)}$ where $d_{in}(v)$ is the in-degree of node v.

The pseudo code of IC model is below:

Algorithm 1 IC_sampling $(graph, seed\_set)$

```
activity_set = seed_set.copy()
active_set = seed_set.copy()
while len(activity_set)> 0 do
    new_activity_set =set()
    for node in activity_set do
        for neighbor in graph[node] do
            if neighbor not in active_set and w((node, neighbor)) <=random.random() then
                active_set.add(neighbor)
                new_activity_set.add(neighbor)
            end if
        end for
    end for
    activity_set = new_activity_set
end while
return len(active_set)
```

Secondly, another popular stochastic diffusion model is named LT(Linear Threshold) model. The model is quite different from IC model in terms of the focus of spreading transition.

The process of spreading of LT model is Algorithm 2.

- In the beginning, each node v selects a random threshold θ_v uniformly at random in range $[0, 1]$.
- If round $t \geq 1$, an inactive node v becomes activated if $\Sigma_{activated\ neighbors\ u} w((u, v)) \geq \theta_v$.
- The weight of the edge (u, v) is calculated as $w((u, v)) = \frac{1}{d_{in}(v)}$ where $d_{in}(v)$ denotes the in-degree of node v.

The pseudo code of LT model is below:

Algorithm 2 LT_sampling $(graph, seed\_set)$

```
activity_set = seed_set.copy()
active_set = seed_set.copy()
threshold = [random.random() for i in range(n)]
while len(activity_set)> 0 do
    new_activity_set =set()
    for node in activity_set do
        for neighbor in graph[node] do
            if neighbor not in active_set then
                w_total=sum(w((neighbor, neighbor_p)) for neighbor_p in graph[neighbor])
                if w_total ≥ threshold[neighbor] then
                    active_set.add(neighbor)
                    new_activity_set.add(neighbor)
                end if
            end if
        end for
    end for
    activity_set = new_activity_set
end while
return len(active_set)
```

Then we have fully understood the two stochastic diffusion models.

Now we are required to consider how to find the seed sets of graphs, which is the core and significant problem of IMP. As a matter of fact, there are so large quantities of algorithms raised since the IMP has been raised up. Among all of them, the IMM algorithm has been the most efficient and accurate algorithm up to now.

IMM algorithm bases on an algorithm which is named TIM. TIM algorithm is a heuristic algorithm and its key idea is to make the seed set intersect with as many RR sets as possible.

The definition of RR(reverse reachable) set is that $\forall u \in RR(v), u \in active\_set(v)$ for a large probability where $u, v \in V$ and $active\_set(u)$ is the active set when the seed set contains only a single node u.

Now we need to consider how to find RR sets. Obviously, the RR sets of IC model can be simply calculated by running the Algorithm 1 with reversed graph $\overline{G}$ and seed set u where u is a random selected node of the n nodes. However, for LT model, things are a little bit different because the spreading of LT model is not symmetric and then cannot be reversed. But since we can find the RR sets of LT model by reversing the spreading process by ourselves, we can easily get the RR sets of any nodes.

After finding the RR sets, we can just use the Algorithm 3 to find the seed set we want.

$F_R(S)$ is the fraction of RR sets which intersect S. The greedy algorithm is meant to make the $F_R(s)$ in the end the maximum.

Algorithm 3 node_selection (R, k)

$S = \emptyset$
for i in range$(1, k)$ **do**
  v =max$_v$(F$_R(S \cup \{v\}) - F_R(S)$ for v in range(n))
  $S = S \cup \{v\}$
end for

The rest part of IMM algorithm is based on the mathematical analysis which uses the concept - martingale. The whole mathematical proof can be found in the paper [2] and will not be included in this report.

In brief, we can combine the Algorithm 3, Algorithm 4 and Algorithm 5 to implement the whole process of IMM algorithm.

Algorithm 4 Sampling (G, k, ϵ, l)

$R = \emptyset$
$LB = 1$
$\epsilon' = \sqrt{2}\epsilon$
$\lambda' = \frac{(2+\frac{2}{3}\epsilon')(\log\binom{n}{k} + l\log n + \log\log_2 n)n}{\epsilon'^2}$
for i in range$(1, \log_2 n - 1)$ **do**
  $x = \frac{n}{2^i}$
  $\theta_i = \frac{\lambda'}{x}$
  while $|R| \leq \theta_i$ **do**
    v =random.randInt$(1, n)$
    R.add$(RR_set(v))$
  end while
  S_i =node_selection(R)
  if $n \times F_R(S_i) \geq (1 + \epsilon') \times x$ **then**
    break
  end if
end for
$\alpha = l\log n + \log 2$
$\beta = \sqrt{(1 - \frac{1}{m})(\log\binom{n}{k} + l\log n + \log 2)}$
$\lambda^* = 2n((1 - \frac{1}{m})\alpha + \beta)^2\epsilon^{-2}$
$\theta = \frac{\lambda^*}{LB}$
while $|R| \leq \theta$ **do**
  v =random.randInt$(1, n)$
  R.add$(RR_set(v))$
end while
return R

Algorithm 5 IMM (G, k, ϵ, l)

$l = (1 + \frac{\log 2}{\log n})l$
$R = Sampling(G, k, \epsilon, l)$
return $node\_selection(R, k)$

3. Empirical Verification

3.1. Dataset

We use three datasets from others, including two datasets from sakai.com which are "network.txt" and "NetHEPT.txt". Another dataset is one from the Internet which is named "twitter-d.txt" which is generated to analyze the social networks on Twitter.

Moreover, in order to guarantee the time limit will not be exceeded and the accuracy of the result will be high, we generate some random data by ourselves. The scales of these datasets are in range $[10^3, 5 \times 10^4]$.

3.2. Performance measure

There are two criteria which can be used to measure the performance of our algorithm. They are:

- *Running time of IMP algorithm*
- *Evaluated influence value from ISE algorithms*

The two criteria are very difficult to distinguish between "good" and "bad" mainly because it is impossible for us to find the exact answer to IMP. But we tried our best to get enough data to guarantee the distinctions.

3.3. Hyperparameters

The ISE algorithms has only one parameter - times for which they run. Because running ISE algorithms once is not enough to estimate the influence of the seed set of network graph, we need to run them many times and calculate the average influence value to make the result more believable. So the value of times for which the algorithms run make sense.

The IMP algorithm IMM has two parameters - ϵ and l. The value of l from the paper [2] is a scientific value which is guaranteed by mathematical proof and experiments, so we had better not edit it.

As a consequence, the only parameter we can edit is ϵ. The meaning of the parameter is that we can guarantee the influence value of our result in the range $[(1 - \frac{1}{m} - \epsilon)best, best]$ where $best$ is the accurate answer to IMP and m is the number of edges.

According to experience, ϵ is often set as 0.1.

3.4. Experimental results

Dataset	Running Time (s)	Influence
network-5-IC	1.112	30.6324
network-5-LT	1.321	34.5412
NetHEPT-5-IC	5.805	323.4887
NetHEPT-5-LT	5.890	392.975
NetHEPT-50-IC	9.820	1298.5584
NetHEPT-50-LT	8.181	1701.9953
NetHEPT-500-IC	18.390	4332.1287
NetHEPT-500-LT	20.794	5587.8998

3.5. Conclusion

The experimental results indicates that the accuracy of IMM algorithm is quite high. Moreover, the running times of the program show that the algorithm is efficient enough.

According to [2], the theoretical expected time complexity of IMM algorithm is $O(\frac{(k+l)(n+m)\log n}{\epsilon^2})$ which means the bigger ϵ is, the slower the algorithm is. If we want to get more accurate answer to IMP, we should spend much more time.

The advantage of the algorithm is that we can control the accuracy by ourselves and get much better answers if we are able to tolerate more running time. In fact, this IMM algorithm is not so good when we need to terminate the program on a certain time limit. And this is its greatest disadvantage.

However, the meaning of IMM algorithm to research is much more important than the practical utility of the algorithm to this course. It may not be better than TIM/TIM^+ algorithms when there is a time limit on the online platform, but it do have theoretical and practical value when we want to think more about mathematics, accuracy and the essence of IMP.

In this project, we analyzed IMP and learned many algorithms related to it. The key point of finding relatively accurate answer to IMP is to find RR sets and try to make seed set intersect them and this algorithm is based on [2]. It is likely that in the near future there will be another algorithm which is more efficient and effective to solve the problem. We should keep pursing high performance and accuracy wherever and whoever we are. Keeping the goal in mind and then sit down to think, prove and calculate. Hope is waiting for us in the future.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.

[2] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1539–1554.