Online Influence Maximization in Graphs
CPT_S 553: Graph Theory Final Project

Reet Barik (WSU ID: 11630142)

December 2020

Abstract

This report takes a look at the *Online Influence Maximization* problem in graphs (social networks in particular) which attempts to identify the best set of "influencers" in a social network to expose a product to such that the set of users becoming aware of it is maximized. This is set in the context of the previously studied *Influence Maximization* problem which is a special case of the more general OIM problem. What follows is report on the development of the OIM problem from IM, some of the common strategies used to solve it, the current state-of-the-art and the future direction and scope of this problem.

1 Motivation

Social media has become a mainstay in the lives of the general public over the last decade or so. Hence, social networks have become one of the main fields where companies try and advertise their product or services. A common marketing campaign generally involves a company giving out free samples of their product to a select few 'influential' nodes (users) on a social network and hope the word-of-mouth from these users convinces their followers or friends to buy it. In such examples, the companies generally operate with a fixed budget (they don't have unlimited free samples of their product to give out). The problem here becomes that of identifying the set of *seed* nodes representing the set of influential users such that when they are influenced, the final influence spread in the social network is maximal.

The above problem has been formalised and referred to as *Influence Maximization* (IM) in the literature. This problem assumes that for every edge in the social network from node u to v, the corresponding edge probability (the probability of u influencing v when u has already been influenced) is known. But this assumption of the edge probabilities being known is not realistic in practice. Given a social network, the edge probabilities are not explicitly available. In the absence of complete information about the influence probabilities, the problem of identifying the seed set of influential users of a given cardinality, such that when influenced or *activated*, results in the influence spread at the end being maximal, is referred to as the *Online Influence Maximzation* problem. Solving the 'OIM' problem makes more sense from the marketer's/advertiser's point of view since this problem formulation is more in sync with the real world setting of social media marketing campaigns.

Marketing on social media is not the only application where OIM can be relevant. With the world coming to a halt due to the unprecedented ramifications of the COVID-19 pandemic, the field of epidemiology is also one that is looking at OIM to combat the rapid spread of diseases. Though the fundamental mechanism may differ, from the infectious disease spread standpoint, the OIM problem, at a high level becomes that of identifying nodes in a network that when isolated or vaccinated results in the final spread of the disease in the population being minimal. Available solutions to OIM cannot be however, treated as a black box solution in this context for reasons addresses in the later sections of this document.

2 Definitions and Technical Background

In the IM literature and subsequently the OIM literature, the spread of the influence has been known to follow either of the following two diffusion process:

- **Linear Threshold Model**: The Linear Threshold model is as follows:
  - Every node v has a threshold value θ_v which is sampled from $U[0,1]$.
  - A node v is influenced by each neighbor u according to a weight $w_{v,u}$ such that

    $$\sum_{w \in neighbor(v)} w_{v,u} \leq 1$$

  - If θ_v fraction of node v's neighbors are active, then v itself becomes active.

  This model describes a very simple and easy to understand diffusion process. This makes it easier to implement/simulate and also makes the underlying maths and corresponding proofs easier. But it fails to capture the influence dynamics of a real-world social network. A more realistic model which is more widely used as a surrogate to simulate real-world influence diffusion in social networks is described below.

- **Independent Cascade Model**: The Independent Cascade model is as follows:
  - Initially, an initial set of nodes S are *activated* or *influenced*.
  - Each edge $e \in E$ corresponding to every (i,j) node pairs that are adjacent in the graph has an associated influence probability p_{ij} (probability of node i influencing j).
  - Every active node v has a one-shot chance of activating a neighbor u with probability p_{vu}.
  - The diffusion process stops when there are no more one-shot activations possible. [Note: If a node u has two active neighbors v and w trying to activate it, it doesn't matter which activates u first.]

  The model described above is a more realistic representation of the dynamics of the real-world process diffusion process and has been adopted as the de-facto model for influence diffusion unless otherwise mentioned.

Given a diffusion model, the Influence Maximization problem takes as input the graph $G = (V, E, p)$, where $v \in V$ are the users or nodes in the social network and $e \in E$ are the edges in between nodes which are connected (friends or followers). p represents the activation or influence probability of every (i, j) pair of vertices such that i and j are adjacent to each other in the graph G $(p_{i,j} \in [0,1])$. The *expected influence spread* σ_S for a node set S is the expected final set of activated nodes in the graph when all the nodes in S are activated initially. The IM problem also takes a parameter k as input which is user-defined. Given the definitions, the mathematical formulation is as follows:
Given a graph $G(V, E, p)$ and a budget k where $k << |V|$, identify the seed set S where $|S| \leq k$, such that upon activating the nodes in S, σ_S is maximal.
This is essentially a constrained optimization problem parameterized by the budget k which puts a constraint on the cardinality of the set S. This makes σ_S the function to be optimized. With this in mind, following are some of the known/proven properties of σ_S:

- The expected influence spread is non-negative
- It is a monotonic function since $\sigma_{S+v} \geq \sigma_S$
- It is sub-modular:
  - Let N be a finite set
  - $\forall S \subset T \subset N, \forall v \in N \backslash T \iff \sigma_{S+v} - \sigma_S \geq \sigma_{T+v} - \sigma_T$
- For a sub-modular, non-negative function which is also monotonic, the task of identifying a k-element set S for which the function is maximized, is NP-hard. This was proved by Kempe *et al.* [2].
- A *greedy algorithm* is a very common solution to such problems, wherein, nodes are added to S one by one in a greedy fashion such that at each step $\sigma_{S+v} - \sigma_S$ is maximum. This gives an $(1 - 1/e)$-approximate solution of the IM problem. In other words, the resulting set S from the greedy algorithm activates at least $(1 - 1/e)$ of nodes that any k-sized set would activate.

The Online Influence Maximization problem is a version of the Influence Maximization problem where the edge probabilities are not known. This calls for repeated interactions with the social network to somehow (use the feedback) learn the edge probabilities directly or some surrogate model representing those probability values. The feedback that can be obtained are of the following kinds:

- Node-level feedback: Here, every interaction or trial (a chosen seed set is activated and the diffusion process is allowed to unfold) is conducted and the final influence spread (set of nodes that got activated) is observed.
- Edge-level feedback: Here, every activation attempt from a node u to its neighbor v is observed and the result (whether it was a successful activation or not) is recorded.

3 Common Solutions and Strategies

Since the focus of this report is on OIM, this section will take a look at the commonly used approaches to solve the OIM problem. The prevalent strategies that are used are as follows:

- **Heuristic Based**: This type of approach is a direct adaptation of the 'greedy' strategy that is used to approximately solve the IM problem. It greedily adds nodes to the set S based on some heuristic till the cardinality of S becomes k. Such approaches are very primitive and do not take into account the influence probabilities on the edges that need to be learned. As a result the solutions obtained are poor in quality but can be arrived at relatively fast. Some of the commonly used heuristics are as follows:
  - Random: This selects k seed nodes arbitrarily giving all nodes in the graph a fair chance of being selected as one of the seeds.
  - Maximum Degree: This sorts the nodes in decreasing order of their *out-degrees* and selects the top k as the seed nodes.
- **Explore-Exploit Policy Based**: This strategy is predominantly used to solve problems under the incomplete information setting and is a cornerstone of most deep-learning or reinforcement-learning based approaches. The underlying approach here is to find a balance between two policies, namely 'Explore' and 'Exploit'. In Explore, the idea is to select seeds that have not been selected in previous trials and use the feedback from the experiment to update the knowledge about the influence probabilities. The Exploit policy on the other hand selects seeds based on the edge probability knowledge that has been learned till the previous trial and cam be essentially thought of as a policy that maximizes return given the currently available information.

The Explore-Exploit strategy is better suited to the OIM problem setting since it is more flexible in terms of giving the user greater control in finding the balance between explore and exploit. Also, it is tailor-made to tackle problems under the incomplete information setting (here, the incomplete information refers to the unknown edge probabilities) wherein, 'explore' results in information gain while 'exploit' results in maximizing influence based on already learned probability information. For this reason, we take a closer look at the different solutions that are used throughout the OIM literature under the umbrella of the Explore-Exploit strategy. They are as follows:

- **Bayesian inference**: This statistical inference based approach uses Bayes' theorem to update the probability of a hypothesis as more information is learned gradually. What follows is how Bayesian Inference can be used in the OIM setting:
  - Each edge activation can be represented as a boolean random variable and hence can be assumed to have a Bernoulli distribution.
  - The edge probabilities are assumed to be drawn from the probability distribution function of a *Beta distribution* [Beta was the choice of distribution since it is a conjugate prior for the Bernoulli distributions, or more generally, binomial distributions].

- For an edge from node i to j, the random variable of the influence probability P_{ij} has a density function:

  $$f_{P_{ij}}(x) = \frac{x^{\alpha_{ij}-1}(1-x)^{\beta_{ij}-1}}{B(\alpha_{ij},\beta_{ij})}$$

  . This makes the mean $E[P_{ij}] = \frac{\alpha_{ij}}{\alpha_{ij}+\beta_{ij}}$ and the square of standard deviation of the distribution $\sigma^2[P_{ij}] = \frac{\alpha_{ij}\beta_{ij}}{(\alpha_{ij}+\beta_{ij})^2(\alpha_{ij}+\beta_{ij}+1)}$

- With this distribution as the prior, one can use the feedback obtained (which edges got successfully activated as a consequence of activating a particular set of seed nodes) as the knowledge gained during the trials where the chosen strategy is Explore and update the posterior distribution by using Bayes' theorem.

- Various strategies can be used to find a balance between the Explore and Exploit strategies. An ε-greedy one is where the agent chooses to explore with a probability of ε and exploit with a probability of $(1-\varepsilon)$.

- After learning the edge probabilities sufficiently, an already existing offline Influence Maximization solution can be used to identify the node set S.

- **Combinatorial Multi-Arm Bandit**: The multi-arm bandit problem has been around for a while now. The problem setting is one where there are multiple arms on a slot machine with each arm having an unknown reward distribution. The aim is to learn those distributions by repeatedly pulling those arms while simultaneously maximizing the expected reward. In the combinatorial setting, the only difference is that multiple arms can be pulled together which might end up stochastically pulling another set of arms. To maximize the reward (or minimize regret) in such a situation while exploring enough to learn the reward distributions can be intuitively mapped to the OIM problem as follows:

CMAB	Symbol	Mapping to IM
Base arm	i	Edge (u, v)
Reward for arm i in round s	$X_{i,s}$	Status (live / dead) for edge (u, v)
Mean of distribution for arm i	μ_i	Influence probability $p_($u,v$)$
Superarm	A	Union of outgoing edges E_S from nodes in seed set S
No. of times i is triggered in s rounds	$T_{i,s}$	No. of times u becomes active in s diffusions
Reward in round s	r_s	Spread $\bar{\sigma}$ in the s^{th} IM attempt

With the above mapping in mind, at each round, a seed set S is selected with $|S| = k$ which is the same as pulling the corresponding super-arm E_S. One can either select S randomly (Explore) or select it based on the available probability estimates by running an offline IM solver (Exploit). Based on this, the influence diffuses through the network resulting in a set of nodes getting activated. The reward here is $\overline{\sigma}(S)$ which is the number of active nodes at the end of the diffusion process and is hence, a non-linear function of the rewards of the pulled arms. The mean probability estimate vector $\overrightarrow{\mu}$ is then updated based on the feedback mechanism. What follows is the above framework illustrated as an algorithm:

Algorithm 1: CMAB FRAMEWORK FOR IM(Graph $G = (V, E)$, budget k, Feedback mechanism M, Algorithm $\mathcal{A}$)

```
Initialize $\vec{\mu}$ ;
$\forall i$ initialize $T_i = 0$ ;
for $s = 1 \to T$ do
    IS-EXPLOIT is a boolean set by algorithm $\mathcal{A}$ ;
    if IS-EXPLOIT then
        $E_S$ = EXPLOIT($G$,$\vec{\mu}$,$O$,$k$)
    else
        $E_S$ = EXPLORE($G$,$k$)
    Play the superarm $E_S$ and observe the diffusion cascade $c$ ;
    $\vec{\mu}$ = UPDATE($c$,$M$) ;
```

4 Related Works and State-of-the-art

Though a relatively new field, the OIM problem is already a well studied one. Lei *et al.* [4] presented a Bayesian Inference based solution which attempted to estimate each edge weight independently and hence was computationally expensive and didn't scale well. There has also been a lot of work where the OIM problem has been approximately solved in the multi-armed bandit setting with corresponding theoretical regret bounds. Vaswani *et al.* [7] proposed a learning algorithm where the agent observed the influenced nodes but not the edges. Carpentier and Valko [1] gives a minimax optimal algorithm for IM bandits but only consider a local model of influence with a single source and a cascade of influences never happens. In a further constrained setting, Sigla *et al.* [6] studies the problem where there is an additional restriction on which nodes can be chosen at each round. Lagree *et al* [3] extends the problem setting by making some nodes persistent over rounds so that they no longer yield results.

The State-of-the-art is a semantic refinement of an approach that was first presented by Wen *et al.* [8] where the authors assumed that the edge probabilities can be approximately expressed as a dot product of two vectors: one was a feature vector of the edge and the other was the one to be estimated through repeated interaction. This approximate solution using a *Linear generalization* of the edge probabilities was adopted by Wu *et al.* [9]. Like in the previous case, the edge probabilities are again expressed as a dot product of two vectors. But the semantics of those vectors are different. This work decomposes the edge probability $p_e \in [0,1]$ on edge e into two d-dimensional latent factors on the source and destination node making up that edge. i.e.,

$$p_e = \theta_{g_e}^T \beta_{r_e}$$

where g_e and r_e denote the source (giving) node and the destination (receiving) node of edge e respectively. The underlying philosophy is that for an edge (ij), θ_i represents the influence of i and β_j represents the susceptibility of j. These two vectors together are a good surrogate for the edge probability. In this setting, the edge probability estimation needn't be done explicitly. One can learn the two latent factors per node to implicitly estimate p_e. This has an additional effect of reducing the complexity of the approach from being $O(|E|)$ to that of $O(d|V|)$.

5 Future Scope and Directions

Despite the amount of literature already available on the OIM problem, there still remains a lot of avenues that remain unexplored. Some of them are follows:

- **Scalability**: With social networks increasing in size rapidly, there is a need for the OIM algorithms to scale accordingly. One way to do that would be to come up with parallel versions of the algorithms. But this might be tricky for those frameworks that use the feedback from one round to select the seeds in the next.

- **Epidemiology**: Intuitively, spread of infectious disease and the corresponding control using vaccination can be modeled as a network. This is where IM/OIM can have an impact by coming up with vaccination strategies (identify nodes which when infected result in maximal spread of the disease and vaccinate them first). Some preliminary work on this front has been started already by Minutoli *et al.* [5]. This however needs further development because the IM/OIM framework can't be used as-is, mainly due to the following reasons:

  - The typical feedback mechanism is not applicable here since it will take a long time to estimate the effect of vaccination from one round. As a result the graph might have changed by the time there is some actionable insight making it obsolete.
  - The influencing mechanism is much different in disease spread than in social media marketing. As a result, the interaction network of a population under consideration needs to tailor made for this using domain knowledge of medical and epidemiology experts.

- **Dynamic Graphs**: Already available solutions to the problem treat it as a static input. But in order to simulate a more real-world scenario there is a need to come up with solutions that take dynamic graphs as input (where nodes and edges are added and/or deleted from one time step to another).

6 Conclusion

In this report, we took a look at how marketing campaigns on social media made the Online Influence Maximization problem in graphs a very relevant one and how it is an extension of the IM problem. This was followed by a brief description of the common strategies used to approximately solve this problem. Some of the related works mentioned in this report gave a brief review of the spread of research in this field followed by an overview of the solution approach of the current state-of-the-art solution. Finally, this report also took a lot how this topic can be taken forward by listing a few possible new directions of resear.

References

[1] Alexandra Carpentier and Michal Valko. Revealing graph bandits for maximizing local influence. 2016.

[2] David Kempe, Jon Kleinberg, and Éva 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*, pages 137–146, 2003.

[3] Paul Lagrée, Olivier Cappé, Bogdan Cautis, and Silviu Maniu. Effective large-scale online influence maximization. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 937–942. IEEE, 2017.

[4] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 645–654, 2015.

[5] Marco Minutoli, Prathyush Sambaturu, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyanaraman, and Anil Vullikanti. Preempt: scalable epidemic interventions using submodular optimization on multi-gpu systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2020.

[6] Adish Singla, Eric Horvitz, Pushmeet Kohli, Ryen White, and Andreas Krause. Information gathering in networks via active exploration. *arXiv preprint arXiv:1504.06423*, 2015.

[7] Sharan Vaswani, Laks Lakshmanan, Mark Schmidt, et al. Influence maximization with bandits. *arXiv preprint arXiv:1503.00024*, 2015.

[8] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. *Advances in neural information processing systems*, 30:3022–3032, 2017.

[9] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646, 2019.