Learning and Maximizing Influence in Social Networks Under Capacity Constraints

Pritish Chakraborty
Indian Institute of Technology, Bombay

Sayan Ranu
Indian Institute of Technology, Delhi

Krishna Sri Ipsit Mantri
Indian Institute of Technology, Bombay

Abir De
Indian Institute of Technology, Bombay

ABSTRACT

Influence maximization (IM) refers to the problem of finding a subset of nodes in a network through which we could maximize our reach to other nodes in the network. This set is often called the "seed set", and its constituent nodes maximize the social diffusion process. IM has previously been studied in various settings, including under a time deadline, subject to constraints such as that of budget or coverage, and even subject to measures other than the centrality of nodes. The solution approach has generally been to prove that the objective function is submodular, or has a submodular proxy, and thus has a close greedy approximation. In this paper, we explore a variant of the IM problem where we wish to reach out to and maximize the probability of infection of a small subset of bounded capacity K. We show that this problem does not exhibit the same submodular guarantees as the original IM problem, for which we resort to the theory of γ-weakly submodular functions. Subsequently, we develop a greedy algorithm that maximizes our objective despite the lack of submodularity. We also develop a suitable learning model that out-competes baselines on the task of predicting the top-K infected nodes, given a seed set as input.

CCS CONCEPTS

• **Information systems** → **Social networks**; • **Computing methodologies** → *Machine learning*; • **Mathematics of computing** → **Submodular optimization and polymatroids**.

KEYWORDS

influence maximization, gamma-weakly submodular functions, deep submodular functions, graph neural networks

ACM Reference Format:
Pritish Chakraborty, Sayan Ranu, Krishna Sri Ipsit Mantri, and Abir De. 2023. Learning and Maximizing Influence in Social Networks Under Capacity Constraints. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570433

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570433

1 INTRODUCTION

The influence maximization (IM) problem aims to find a set of seed nodes that maximizes the spread of information across a social network [23]. The solution to this problem is central to the viral marketing application where an advertiser searches for a given number of users in a social network to spread information to the largest number of potential customers [2, 6, 9, 11, 33, 36, 39]. At the outset, it has two components, *viz*., modeling of the underlying information diffusion process and designing the optimal seed node selection strategy. Research on influence maximization, from both these perspectives, abounds in literature [1, 3–8, 15, 17, 17, 22–24, 27–29, 34, 37, 43, 44]. These models cast the expected spread of information as a monotone submodular function and resort to variants of greedy maximization algorithm [7, 17, 18, 21, 23, 32, 40].

In general, the existing influence maximization settings aim to spread the information to the largest number of nodes. To this end, they aim to maximize the expected number of users influenced by the seed nodes — the sum of the infection probabilities of *all* the nodes. Our work is motivated by the observation that in several applications, one may like to reach only a specific number of users. For example, consider an advertising campaign for a job role through a social network such as LinkedIn. Often, the hiring farm is limited in the number of candidates it can interview. In this scenario, spreading the advertisement to each user in the social network may be suboptimal in terms of getting best possible responses. Rather, the campaign should be customized towards influencing K users with the highest possible likelihood; K indicates the number of interviews to be conducted. This problem can be cast as an instance of maximizing the sum of top-K infection probabilities of the users, where K indicates the *capacity* of influenced users targeted in the campaign.

In a related setup, observations from social network data may often consist of pairs of {(seed set, infected nodes)} without any additional information, *e.g.*, time or sequence of infection, *etc.*. due to privacy issues. One can think of the set of infected nodes as the nodes that are most likely going to be infected by the seed nodes. In order to learn the influence of a seed set on other nodes, one needs to learn a predictor of top-K nodes that are likely to be infected, given a set of seed nodes.

1.1 Our contributions

Driven by the above motivation, we introduce the influence maximization problem of top-K nodes and develop TopK-InfluMax, a framework for optimal seed node selection specifically catered to solve this problem. Moreover, we develop InfluNet a neural set network, which predicts the top-K nodes for a set of seed nodes in a graph. Specifically, we make the following contributions.

Influence maximization of top-K nodes. We formulate the problem of influence maximization where the goal is to maximize the sum of top-K infection probabilities across the nodes in a network. K indicates the *capacity* of the campaign in terms of number of influenced users desired. Like traditional influence maximization, this too is NP-Hard. However, the proposed formulation changes the theoretical characterization of the influence maximization problem dramatically. Specifically, unlike the original formulation, influence maximization over top-K nodes is neither monotone nor sub-modular, which renders the approximation algorithms based on the iterative greedy approach ineffective. Therefore, we require an algorithm tailored towards the proposed problem, which we address through the proposed framework called TopK-InfluMax. Here, we first show that our objective can be expressed as a difference between a γ-weakly submodular function and a modular function [20]. This characterization allows us to leverage a stochastic distorted greedy algorithm to obtain an approximately optimal set of seed nodes.

Neural set network of top-K node-set prediction. Next, we aim to design InfluNet, a trainable neural set to set network. At the very outset, InfluNet models the infection probability of each node using a nonlinear submodular network on the graph embeddings induced by a graph neural network. In detail, InfluNet consists of a GNN and a message passing neural submodular function (MPNSF). The GNN generates structural embeddings, which are fed into MPNSF, which, in turn, characterizes the influence flow across the network starting from the seed set. In general, an infection probability is a submodular function of the seed nodes. To that aim, we develop a graph-induced neural submodular model that computes a submodular function at each layer by iteratively applying concave functions on submodular functions from the previous layer [10]. Specifically, for each node, the neural submodular model aggregates the infection probabilities of its neighbor, applies a concave function on top of it, and re-applies these two steps for a number of layers. Such an iterative framework captures the effect of influence from a distant node and enforces submodularity in the computation of infection probabilities. Finally, given pairs of seed-set and influenced nodes, *i.e.*, (S, I), we train InfluNet by applying the sigmoid function on the network outputs and feeding them to a Dice coefficient-based loss function. Such a loss function ensures that the sigmoid scores are as close to the ground truth infection labels (0 for un-infected and 1 for infected) as possible.

We experiment with a wide variety of real world datasets, which shows that TopK-InfluMax outperforms several state-of-the-art influence maximization algorithms, by choosing the seed nodes that provide a high value of the sum of top-K infection probabilities for a wide range of capacity values (K). The performance boost starts to gain prominence when the seed set budget becomes moderate. Moreover, we show that our proposed neural model InfluNet is able to outperform other baselines, showing a larger overlap between the predicted set of influenced nodes and the ground truth set of influenced nodes.

2 PRELIMINARIES

In this section, we present a brief overview of the information diffusion models, the existing influence maximization problem and a high level overview of our goal.

2.1 Information diffusion

We are given a directed graph $G = (V, E)$ where a node u can directly influence v if $(u, v) \in E$. We define $N(u) = \{v \mid (v, u) \in E\}$ and $N^{out}(u) = \{v \mid (u, v) \in E\}$ indicating the sets of incoming and outgoing neighbors of u respectively. We also assign q_u as the weight of each node. In the context of job advertisement, q_u may signify the quality of node u.

Information diffusion is a sequential process initiated by a set of seed nodes that act as the source of information. In this process, each node can be in one of the three states at a given time t: (i) *active* (ii) *inactive* and (iii) *susceptible*. Active nodes are those that can influence (activate) other nodes. Specifically, at any time, if a node u receives the information or gets influenced through an incoming neighbor, then it remains active for the next time step. Inactive nodes are influenced nodes but are not active at the current time. Susceptible nodes are those nodes that have not been influenced in the past and are therefore potential candidates to be activated in the future.

Given the seed set S, we denote the set of influenced nodes until time t as $I_S(t)$. Thus, the set of active nodes at time t is $A_S(t) = I_S(t) \backslash I_S(t-1)$. Similarly, the set of inactive nodes at time t is $I_S(t) \backslash A_S(t) = I_S(t-1)$ and the set of susceptible nodes at time t is $V \backslash I_S(t)$. For each time t, a node u can only be influenced by the active set $A_S(t)$. We also denote $I_S = \lim_{t \to \infty} I_S(t)$ as the set of nodes that eventually become influenced. Finally, we define $p(u; S)$ as the infection probability of node u given that S is the seed set, *i.e.*, $p(u; S) = \mathbb{P}(u \in I_S)$. Existing works predominantly use two types of processes for Information diffusion:

(1) **Independent cascade (IC):** It assigns a probability to each edge $(u, v) \in E$ as $p_{uv} \in [0, 1]$. At each time step t, an active node $u \in A_S(t-1)$ influences a subset of its out-neighbors $\mathcal{N} \subset N^{out}(u)$ which have never been influenced *i.e.*, $\mathcal{N} \subseteq N^{out}(u) \backslash I_S(t-1)$ with the edge probability p_{uv}.
(2) **Linear threshold (LT):** In this mode, each edge is assigned a probability $p_{uv} \in [0, 1]$. A node $u \in V \backslash I_S(t-1)$ is influenced at time t if $\sum_{w \in A_s(t-1) \cap N(u)} p_{uw} \geq \tau_w$ for a given threshold τ_w.

2.2 Influence maximization

Given an information diffusion model $\mathcal{M}$, a traditional influence maximization task aims to find the seed nodes S with $|S| \leq b$ that maximizes the expected number of nodes that becomes eventually infected. Specifically, one aims to solve the following problem:

$$\text{maximize}_{S \subset V} \ \mathbb{E}[|I_S|] \quad \text{such that, } |S| \leq b < |V| \tag{1}$$

Here, the expectation is taken over different random realizations of the diffusion process under $\mathcal{M}$.

2.3 Our goal

Our goal in this paper is two-folds. In many applications, *e.g.*, advertisement for niche job applications, one may be interested to ensure the information reaches a limited number of suitable candidates

rather than all the users. To this end, we aim to find seed nodes that will maximize the sum of top-K infection probabilities. On the other hand, in the context of designing predictive models for information diffusion, our goal is to design a predictive model that can accurately predict the set of top-K influenced nodes for any seed set, given a value of K.

3 MAXIMIZING INFLUENCE FOR TOP-K NODES

In this section, we formulate the problem of influence maximization under capacity constraints and then provide a novel characterization of the underlying objective. Finally, we present the stochastic distorted greedy algorithm to solve the problem.

3.1 Problem formulation

Given a graph $G = (V, E)$, the node weights $\boldsymbol{q} = [q_u \mid u \in V]$, our goal is find the set of seed nodes S that maximizes the sum of the top-K weighted probabilities of the influenced nodes; K denotes the capacity constraint. More specifically, we aim to solve the following optimization problem:

$$\underset{S \subset V}{\text{maximize}} \; F_K(S) \quad \text{such that, } |S| \leq b \tag{2}$$

$$\text{with, } F_K(S) = \max_{I \subseteq V \setminus S : |I| \leq K} \sum_{u \in I} q_u \, p(u; S) \tag{3}$$

Recall that, $p(u; S)$ is the probability that node u gets eventually influenced given that S is the seed set, *i.e.*, $p(u; S) = \mathbb{P}(u \in I_S)$. We would like to highlight that in Eq. (3), we aim to find the top-K infected nodes that are not in the seed set, *i.e.*, $I \subseteq V \setminus S$. The above optimization problem assigns the weight q_u to each node u, which encourages that the nodes having high values of q_u are influenced with high likelihood. Such an approach is useful in advertising niche job applications, where one may want the information to be spread to a limited number people, quantified through K, but with high competence.

Hardness of the objective. Like the traditional influence maximization problem (1), solving the optimization problem (2) is also NP-Hard. If we have $q_u = 1$ for all $u \in V$, then for $K = |V|$, we have $F_K(S) = \mathbb{E}[I_S]$, which reduces the optimization problem (2) to the traditional influence maximization problem (1) which is a well-known NP-Hard problem [23].

3.2 Characterization of $F_K(S)$

Here, we provide a novel characterizations of $F_K(S)$ by starting with a counterexample that shows that $F_K(S)$ may not be monotone and submodular in general. Before formally presenting those results, we first state few definitions in this context.

Definition 1. *Given a set function $F : 2^V \to \mathbb{R}$, we define its marginal gain $F(u \mid S) = F(S \cup \{u\}) - F(S)$. (i) The function F is monotone non-decreasing in S if $F(u \mid S) \geq 0$ for all $u \in V \setminus S$. (ii) The function F is γ-weakly submodular if for $T \subset V \setminus S$, we have $\sum_{u \in T} F(u \mid S) \geq \gamma [F(T \mid S)]$ for $\gamma > 0$ [20]. If $\gamma = 1$, F becomes submodular.*

Non-monotonicity and non-submodularity. It is well known that the objective for traditional influence maximization problem (1) is monotone and submodular for IC and LT models. [23] However, our objective (3) for maximizing the weighted sum of top-K influence probabilities does not guarantee either of these properties as noted in the following proposition [1].

Proposition 2. *Assume that the information diffusion process follows an IC model. Then, the function $F_K(S)$ defined in Eq. (3) may be neither monotone nor submodular in S for $K < |V|$.*

Novel representation of $F_K(S)$. Although $F_K(S)$ need not be monotone and submodular, we show that $F_K(S)$ can be represented as a difference between a monotone and γ-weakly submodular and a modular function. Specifically, we express $F_K(S)$ as follows:

$$F_K(S) = \underbrace{\left[\max_{I \subseteq V \setminus S : |I| = K} \sum_{u \in I} q_u \, p(u; S) + \rho |S| \right]}_{H_K(S)} - \underbrace{\rho |S|}_{m(S)} \tag{4}$$

where $\rho > 0$ is a constant. Here, we first add the modular function $m(S) = \rho|S|$ with $F_K(S)$ to obtain $H_K(S)$ and then subtract it. In the next Theorem, we present our result on monotonicity of $H_K(S)$.

Theorem 3. *Let the infection probabilities $p(u; S) < p_{\max} < 1$ for $u \notin S$ and the node weights $q_u < q_{\max}$. Then the set function $H_K(S)$ defined in Eq. (4) is monotone in S if $\rho > q_{\max}$ whenever $p(u; S)$ is monotone submodular.*

Although $H_K(S)$ is monotone, it is not guaranteed to be submodular. This is because, $H_K(S)$ can be expressed as sum of the set function $F_K(S)$ and the modular function $\rho|S|$. Thus, if $H_K(S)$ is submodular, $F_K(S)$ would also have been submodular which is not the case as shown in Proposition 2. However, $H_K(S)$ satisfies γ-weak submodularity as shown in the following theorem.

Theorem 4. *Assume that, for all node $u \in V$, $q_{\min} < q < q_{\max}$ and $0 < p_{\min} < p(u; S) < p_{\max} < 1$ for $u \notin S$. Then, under the conditions of Theorem 3, $H_K(S)$ is γ-weakly submodular with $\gamma > \gamma^*$ where γ^* is given by:*

$$\gamma^* = \max \left\{ \frac{q_{\min} p_{\min}}{\rho(p_{\min} + 1) - q_{\min} + 2(|V| - K)\rho p_{\max}}, \frac{q_{\min} p_{\min}}{\rho + K p_{\max}} \right\}. \tag{5}$$

Here, in Eq. (5) the two values of γ^* within $\max\{\cdot, \cdot\}$ indicate the bound of γ on two different regimes. The first term within max operator indicates the bound for higher values of K. Specifically, when $K \to |V|$, the first term becomes $\frac{q_{\min} p_{\min}}{\rho(p_{\min}+1) - q_{\min}}$. Moreover, when $q_u = \rho$ for all $u \in V$, this ratio goes to unity— it reduces to the result on submodularity of the objective of the existing influence maximization problem. The second term $\gamma > q_{\min} p_{\min} / (\rho + K p_{\max})$ captures the bound for low values of K. Note that, in general computing both $p_{\max}$ and $p_{\min}$ can be difficult. However, they can be easily estimated using *Monte Carlo* simulations for any IC or LT model.

TopK-InfluMax algorithm. Algorithm 1, summarizes TopK-InfluMax which maximizes F_K (2). Based on the equivalent formulation of (4), we use the *stochastic distorted greedy* algorithm [20] to maximize the difference between a monotone γ-weak submodular function and a modular function, with a *distortion factor* set to 1. In each iteration, we sample a subset B_i of nodes from the graph, where the cardinality of B_i is taken to be the size of

[1] Proofs of all technical results are in the Appendix, in the supplementary material.

Algorithm 1: TopK-InfluMax [20]

Require: Budget b, the value of ρ, edge probabilities $\{p_{uv}\}$ $\forall u, v \in V$, a fraction $s \in [0, 1]$.

$S \leftarrow \emptyset$
for $i \in [b]$ **do**
  Randomly draw a subset of nodes B_i of size $s|V|$.
  EstimateInfectionProbs($G, \{p_{uv}\}_{(u,v)\in E}$)
  $v^* \leftarrow \text{argmax}_{v \in B_i} H_K(v|S) - \rho$
  if $\left(1 - \frac{\gamma}{b}\right)^{|V|-i} H_K(v^*|S) - \rho > 0$ **then**
    $S \leftarrow S \cup \{v^*\}$
Return S

approximated minimum weighted vertex cover. In practice, we substitute the min-weighted vertex cover number with 75% of the size of V. We include the node with the highest positive marginal gain in the solution set S in each iteration.

Note that EstimateInfectionProbs finds the infection probabilities for each node pair by simulating the information cascade process $\mathcal{M}$ on edge weights $\{p_{uv}\}$. It is sufficient to simulate the IC model [23]. We employ MC simulations to collect the total number of times a node is "active" or has been influenced after all diffusion MC rounds have completed, and then divide by the total number of rounds. In practice, the cascade runner process is a subroutine of F_K.

Finally, we quote the approximation guarantee of Algorithm 1, which is proved in [20].

Theorem 5. *If s in Algorithm 1 is given by $s = \lceil \frac{n}{k} \log\left(\frac{1}{\epsilon}\right) \rceil$, then Algorithm 1 returns a set S which satisfies*

$$\mathbb{E}[F_K(S)] \geq (1 - \exp(-\gamma))H_K(OPT) - \rho(OPT) \tag{6}$$

4 DESIGN OF INFLUNET

So far, we have assumed that the diffusion process follows IC or LT. Realistically, the mechanics of the diffusion process needs to be *learned* from data. In this section, we achieve this objective. Specifically, we formulate the problem of *learning* an influence model from {seed set, infected nodes} that outputs the set of top-K infected nodes I for a given seed set S.

To learn the diffusion process, we design InfluNet, which takes a seed set S as input and models the influence probability scores $p_\theta(u; S)$ for each node $u \in V$, where these scores are modeled using neural network parameterized by θ. Therefore, instead of assuming a fixed parameterized model, *e.g.*, IC or LT model, we rely on expressive neural networks to uncover the underlying information diffusion process.

4.1 Overview of InfluNet

Irrespective of modeling choices, the probability $p(u; S)$ is a monotone function in S. Moreover, in practice, it should also follow the law of diminishing returns. To this aim, we approximate $p(u; S)$ using a *neural submodular* function $p_\theta(u; S)$, which is parameterized using θ. In a nutshell, it is a message passing network, which recursively computes submodular functions at each node u by applying a *concave map* on the submodular functions obtained in the previous step [10].

For each node u, we maintain a probability score $p^{(n)}(u; S)$ at each step of recursion n. Then, at step $n+1$, we first gather $p^{(n)}(v; S)$ from all $v \in N(u)$— the probability scores of the neighbors of u computed at the previous step n— and feed each of them into a concave function. Next, we add the outputs from the concave function with a modular function and finally pass the sum through another concave function to obtain the $p^{(n+1)}(v; S)$

4.2 Neural Parameterization

InfluNet consists of two components (i) a message passing neural submodular function (MPNSF) and (ii) a graph neural network (GNN).

Message passing neural submodular function. Given a concave function $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ and a submodular function $f(S)$, it is well known that $\phi(f(S))$ is submodular [12, 13, 31]. We leverage the idea to build a recursive neural model of monotone submodular function similar to [10]. Given a graph $G = (V, E)$, a set of node embeddings $\{\boldsymbol{x}_u \mid u \in V\}$, a seed set $S \subset V$ an integer N, we feed the probability scores $p^{(n)}(v; S)$ of the neighbors of u into a concave function ϕ. Next, we aggregate $\phi(p^{(n)}(v; S))$ from all in-neighbors $v \in N(u)$ using attention-weights α_{uv} and add to it a modular function $m_\theta^{(n)}$. Finally, we feed the output into another concave function ψ to update $p^{(n)}(u; S) \to p^{(n+1)}(u; S)$. We formally represent this formulation as follows:

$$p^{(n)}(v \mid S) = 1 \qquad \text{for } n \in [N],\ v \in S \tag{7}$$

$$p^{(0)}(v \mid S) = 0 \qquad \text{for } v \notin S \tag{8}$$

$$p^{(n+1)}(u \mid S) = \psi\Bigg(\sum_{v \in N(u)} \alpha_{uv} \phi(p^{(n)}(v \mid S)) + \sum_{v \in S \cup N(u)} m_\theta^{(n)}(\boldsymbol{x}_v) \Bigg) \qquad \text{for } n \in [N],\ v \notin S \tag{9}$$

$$p_\theta(u; S) = p_\theta^{(N+1)}(u; S) \tag{10}$$

where, α_{uv} are computed in the following manner:

$$\alpha_{uv} = \frac{e^{\gamma_\theta(\boldsymbol{x}_u, \boldsymbol{x}_v)}}{\sum_{v' \in N(u)} e^{\gamma_\theta(\boldsymbol{x}_{v'}, \boldsymbol{x}_u)}} \tag{11}$$

Here, note that ϕ and ψ are concave functions and both are in $[0, 1]$; $m^{(\bullet)}$ is a positive neural network that consists of a linear and ReLU layer; and, γ_θ is neural network that comprises of a linear, ReLU and linear networks. Note that Eq. (7) ensures that the probability scores for each node of the seed set S is always 1. This is in line with the real-world interpretation that seed nodes are already infected at the start. The recursive update (9) follows a message passing protocol where a node acquires messages in the form of submodular functions from its neighbors. Note that $p_\theta(u; S)$ is monotone submodular for all nodes $u \in V$ as formalized by the following proposition.

Proposition 6. *Given that the functions ϕ and ψ are positive, monotonically increasing and concave; and, the function $m_\theta^{(n)}$ is a positive. Then, $p_\theta(u; S)$ is a monotone submodular set function in S for all $u \in V$.*

Graph neural network for computing $\{\boldsymbol{x}_u\}$. We compute the representations $\{\boldsymbol{x}_u\}$ using a graph neural network (GNN) [14].

Dataset	$\|V\|$	$\|E\|$	d_{avg}	Diameter
Deezer	28,281	92,752	3.27	21
CM Physics Citation	23,133	93,497	4.04	14
GNUtella 1	26,518	65,369	2.46	10
GNUtella 2	22,687	54,705	2.41	10

Table 1: Dataset statistics for experiments with capacity constrained influence maximization.

Here, we first start with a feature encoder $\textsc{FeatEnc}_\theta$ which takes the node features $\boldsymbol{f}_u$ as input and output an initial embedding $\boldsymbol{x}_u(0)$, *i.e.*,

$$\boldsymbol{x}_u(0) = \textsc{FeatEnc}_\theta(\boldsymbol{f}_u) \quad \text{for all } u \in V \tag{12}$$

Then, given an integer L, GNN first gathers information from $\ell < L$ hop neighbors using L propagation layers, then combine them with a symmetric aggregator and finally update the node representations in the following recursive manner.

$$\boldsymbol{z}_{(u,v)}(\ell-1) = Z_\theta(\boldsymbol{x}_u(\ell-1), \boldsymbol{x}_v(\ell-1)) \tag{13}$$

$$\boldsymbol{x}_u(\ell) = \Lambda_\theta\left(\boldsymbol{z}_{(u,v)}(\ell-1), \textstyle\sum_{v\in N(u)} \boldsymbol{x}_v(\ell-1)\right) \tag{14}$$

Finally, we compute the embedding of node u as $\boldsymbol{x}_u = \boldsymbol{x}_u(L)$. The parameters of ϕ, ψ, $m^{(n)}$ and γ_θ in Eqs. (7)– (10) as well as $\textsc{FeatEnc}$, Z and Λ are decoupled, and are collectively denoted as θ.

4.3 Model training

Given a set of $\{(S_i, I_i) \mid i \in [M]\}$ where S_i is a seed set of nodes and I_i are the corresponding infected nodes in the same graph $G = (V, E)$, our goal is to estimate the model parameters θ which would predict the infected set $\hat{I}$ that is closest to the ground truth infected set I for a given seed set S. Hence, we aim to maximize the sum of the Jaccard coefficients between I_i and $\hat{I}_i$, *i.e.*, $\sum_{i\in[M]} |I_i \cap \hat{I}_i|/|I_i \cup \hat{I}_i|$. However, it is not differentiable. Thus, we resort to optimizing a differentiable surrogate of Jaccard coefficient, given as follows.

$$\underset{\theta}{\text{maximize}} \quad \sum_{i\in[M]} \frac{\sum_{u\in I_i} p_\theta(u \mid S_i)}{\sum_{u\in I_i} p_\theta(u \mid S_i) + |I_i|} \tag{15}$$

In practice, we also add a small smoothing factor s to the numerator and denominator of the above objective, which helps in better training. The above function is also called Sorensen-dice coefficient [41].

Once trained, we can return the nodes having highest top-K values of $p_\theta(u; S)$ for an unseen seed set S.

5 EXPERIMENTS ON MAXIMIZING INFLUENCE

In this section, we evaluate our algorithm TopK-InfluMax which maximizes the sum of top-K weighted probabilities, on four diverse datasets against three competitive baselines.

5.1 Experimental Setup

Datasets. We run experiments on four public datasets, *viz.*, Deezer Social Network (Deezer), two Gnutella Network (Gnutella-1, Gnutella-2) and Conditional Matter Physics Citation Network (CA-condmat). For each dataset, we consider IC model with different edge probabilities. The exact generative mechanism of edge probabilities varies across experiments and is mentioned therein. For each dataset, we generate q_u uniformly at random. We summarize the datasets in Table 1.

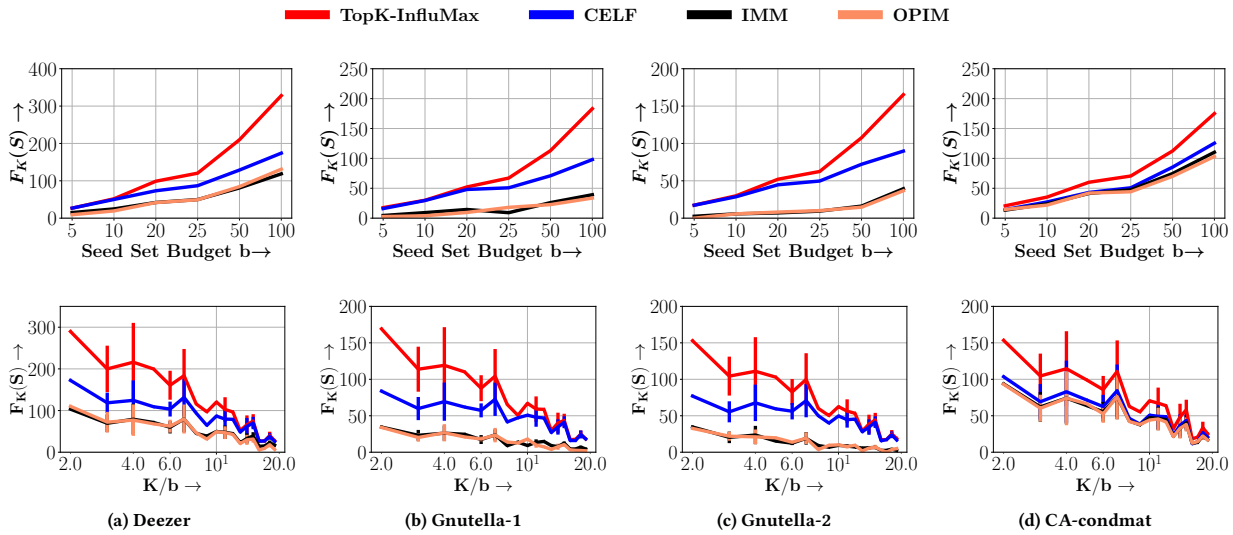
Baselines. We compare TopK-InfluMax against three state-of-the-art influence maximization algorithms, *viz.*, (i) Cost Effective Lazy Forward (CELF) [16, 29], (ii) Influence Maximization with Martingales (IMM) [43] and (iii) Online Processing algorithm for IM [42] (OPIM) . All these algorithms were designed to find the seed set S which would maximize the expected influence spread across the entire network. IMM is run with $\epsilon = 0.1$, while OPIM is run with a sample size of 1000 reverse reachable sets.

Evaluation protocol. Given a graph $G = (V, E)$, the underlying edge probabilities $\{p_{uv} \mid (u, v) \in E\}$, the desired size of the infected set K and the seed set budget b, we run the baselines and TopK-InfluMax on the graph, which provides us a solution S for the optimal seed set with $|S| \le b$. Since the baselines optimize a monotone submodular function the seed set they output always has size b. However, our objective function $F_K(S)$ described in Eq. (3) is neither monotone nor submodular as suggested by Proposition 2. Finally, we evaluate the quality of all the methods using $F_K(S)$, the sum of top-K probabilities. To compute this quantity, one needs to compute $p(u; S)$. To do so, we perform 1000 rounds of Monte Carlo simulations and approximate $p(u; S)$ using the fraction of times u gets infected given the set S.

5.2 Results

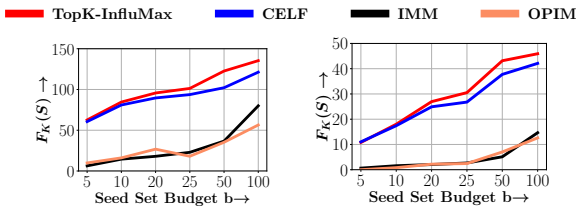
Performance comparison. We first compare our method against the three baselines, *i.e.*, CELF [29], IMM [43] and OPIM [42]. Here, we set the edge probability as $p_{uv} = 1/|N^{out}(u)|$ where $N^{out}(u)$ is the out-degree of u. Figure 1 summarizes the results in terms of variation of $F_K(S)$ vs b (top-row) and $F_K(S)$ vs K/b, *i.e.*, the ratio of the number of nodes selected to the budget (we vary both K and b). We make the following observations. (i) TopK-InfluMax outperforms the baselines across the entire horizon of K and b. (ii) CELF [29] outperforms all other baselines. CELF employs a lazy forward algorithm where the marginal gain of the objective function is selectively computed. However, it does not perform any sampling. In contrast, IMM and OPIM are both based on sampling, where paths and nodes are drawn during the computation of the reverse-reachable-set, which is the core data structure used in IMM and OPIM. As a result, CELF performs much better. (iii) As b increases, the performance boost of TopK-InfluMax against the baselines becomes more and more prominent. This is because as the size of S increases, $p(u; S)$ across all nodes u become high. Hence, the value of $p_{\min}$ as well as γ^* for the function $H_K(S)$ (see Theorem 4) becomes high. This renders an improved approximation factor $1 - \exp(-\gamma)$ in our algorithm. (iv) In the context of variation of performance with K/b, we observe that TopK-InfluMax admits a competitive advantage for low values of K/b, whereas for a high value of K/b, the difference between TopK-InfluMax and CELF gets smaller. This is because as K/b increases $F_K(S)$ goes close to the objective of traditional influence maximization problem which maximizes the spread of influence across all possible nodes.

Next, we consider two other distributions of edge probabilities— (1) Constant where $p_{uv} = 0.2$ and (2) Trivalent where $p_{uv} \sim \text{Unif}\{0.1, 0.01, 0.001\}$ and compare TopK-InfluMax against the



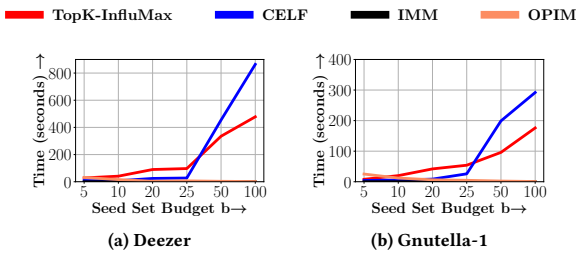
(a) Deezer (b) Gnutella-1 (c) Gnutella-2 (d) CA-condmat

Figure 1: Variation of $F_K(S)$, *i.e.*, sum of top-K weighted probabilities against seed set budget b (top row) and the value of the ratio K/b (bottom row), for TopK-InfluMax and all the baselines, *viz.*, CELF [29], IMM [43] and OPIM [42] across all four datasets. In all cases, the edge probability is set as the inverse out degree of the source node, *i.e.*, $p_{uv} = 1/|N^{out}(u)|$. For the first row, $K = 400$. We observe that TopK-InfluMax outperforms the baselines for a wide landscape of both K and b.



(a) Constant (b) Trivalent

Figure 2: Variation of $F_K(S)$, *i.e.*, sum of top-K weighted probabilities vs seed set budget b when $p_{uv} = 0.2$ (Constant, panel (a)) and $p_{uv} \sim$ Unif $\{0.1, 0.01, 0.001\}$ (Trivalent, panel (b)) using TopK-InfluMax and all the baselines, *viz.*, CELF [29], IMM [43] and OPIM [42] on Gnutella-2. In each case, $K = 400$.



(a) Deezer (b) Gnutella-1

Figure 3: Execution time of different algorithms for different value of budget b with $K = 300$, for TopK-InfluMax and all baselines on the Deezer and Gnutella-1 networks.

three baselines, *i.e.*, CELF [29], IMM [43] and OPIM [42]. Figure 2 summarizes the results. We make the following observations. (i) Our method outperforms the baselines; (ii) The difference between our method and CELF is smaller than the difference when $p_{uv} = 1/N^{out}(u)$. This is because the structural dependence on the infection probabilities for both trivalent and constant are significantly lower than the degree based edge probability computation. Thus the variance of infection of probabilities across the nodes are very low. As a result, the top-K nodes found by CELF also admit similar value of $F_K(S)$ as TopK-InfluMax.

Dataset	$\|V\|$	$\|E\|$	$\|\{(S_i, I_i)\}\|$	$\mathbb{E}[\|I_i\|]$	$\mathbb{E}[S_i]$
Digg [38]	30,268	519,250	1421	115.738	22.178
Weibo [38]	29,378	1,127,024	2000	3.459	2.0
Cit-HepPh [30]	34,546	421,578	1000	128.912	120.067

Table 2: Datasets for evaluation of InfluNet.

Analysis of running time. Next, we compare the running times of TopK-InfluMax against the baselines. Figure 3 summarizes the results. While TopK-InfluMax is slower than IMM and OPIM, it is still tractable and faster than CELF for high values of b. Therefore, given the large size of the graphs and the performance it achieves, it is still practically very useful despite high latency. It would be interesting to extend similar path and node sampling approaches as IMM and OPIM to our setup. However, sampling may lower efficacy as already visible Fig. 1.

6 EXPERIMENTS ON LEARNING INFLUENCE MODEL FROM (S, I) PAIRS

In this section, we evaluate InfluNet against five competitors on three datasets to show that InfluNet learns the influenced nodes from the seed sets with higher accuracy than baselines.

6.1 Experimental Setup

Machine Setup. We performed our neural experiments on a 64-core Intel Xeon Silver 4216 CPU (2.10GHz) server running Ubuntu 20.04.3 LTS, and provisioned with two NVIDIA RTX A6000 GPUs and 1TB of RAM.

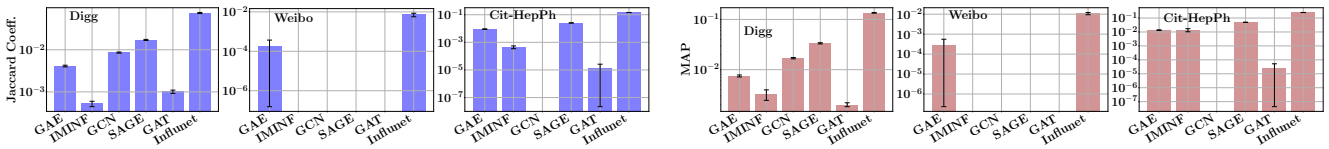


Figure 4: Prediction of infected nodes from $\{(S_i, I_i)\}$. Performance measured in terms of mean Jaccard Coefficient (left three panels) and mean average precision (MAP, right three panels) for InfluNet and all the baselines, *i.e.*, GAE [26], IMINF [38], GCN [25], SAGE [19] and GAT [45] for Digg, Weibo and Cit-HepPh datasets.

Datasets. We use the three datasets listed in Table 2. While Digg and Weibo contain real-world cascade data, Cit-HepPh does not. Thus, we create synthetic cascades in Cit-HepPh by performing 1000 rounds of MC simulations under IC with trivalent edge weights, wherein edge weights are randomly sampled from $\{0.001, 0.01, 0.1\}$. Cascades provide us ground-truth (S, I) pairs, where S corresponds to the set of nodes starting a cascade and I corresponds to the first K nodes getting infected.

Baselines. We compare InfluNet against five baselines: (1) *IMINF*: We train IMINFECTOR [38], which is an algorithm designed for learning influence probabilities from cascades, on our datasets. Using the diffusion matrix output by the model, we perform inference to obtain $\hat{I}$. (2) *GAE*: We train a graph autoencoder [26] on the link prediction task. We infer edge weights for the given graph using the trained autoencoder and using these edge weights. During test, we perform MC simulations using an IC model for a given seed set S to obtain top-K infected nodes $\hat{I}$. (3)*GCN*: We train the Graph Convolutional Network (GCN) [25] using a binary node classification task where $y_v = +1(-1)$ if v is (not) infected. Given a seed set S and a node v, we model the classifier using a neural network as $y_v = \text{NN}(\boldsymbol{x}_v, \sum_{u \in S} \boldsymbol{x}_u)$. Here NN is a deep neural network which consists of a linear (128×128), ReLU and another linear layer (128×1). (4) *SAGE*: Here we use another GNN, *viz.*, GraphSAGE [19] and build the predictor of infected nodes using the same method as GCN. (5) *GAT*: Here we use another GNN, *viz.*, GAT [45] and build the predictor of infected nodes using the same method as GCN. For the baselines (3–5), we use the final classifier NN to output a score $\hat{y}_v$ for every node for an test seed set S. Based on these scores, we obtain top-K nodes.

Evaluation protocol. We split the datasets of $\{(S_i, I_i)\}$ pairs into 60% training, 10% validation and 30% test folds. We then present our training and validation folds before all the models for training. Once the model is the trained, we provide a test seed set S_i to the trained model which in turn returns a ranking of all nodes R in terms of their infection probability or score. In order to measure how does this ranking comply with the gold infected nodes I_i, we use two metrics: (i) the Jaccard coefficient between between I_i and the top $|I_i|$ nodes in R, *i.e.*, $I_i \cap R[1...|I_i|]/I_i \cup R[1...|I_i|]$ and the average precision computed [35] as $Avp_i = \sum_{j=1}^{|V|} \frac{\mathbf{1}[R[j] \in I_i]|R[1...j] \cap I_i|}{|I_i|}$ where $\mathbf{1}[\cdot]$ is an indicator function. Finally, we compute the average over all pairs of $\{(S_i, I_i)\}$ and report mean Jaccard coefficient and mean average precision (MAP).

Implementation details. In all the experiments, we use $\dim(\boldsymbol{x}_\bullet) = 16$ and $\phi(x) = \psi(x) = 1 - e^{-x}$ (9). We train each model for 30 epochs and select the model as the one which has given the best value of mean Jaccard coefficient on the validation set, across 30 epochs. Additional details and code are in https://gitlab.com/PritishC/influence-maximization.

6.2 Results

Figure 4 shows that InfluNet outperforms all the baselines by a significant margin. While InfluNet is specifically engineered towards uncovering the sequential diffusion process, none of the baselines can achieve this task. Hence, mapping the task to simpler problems such as node classification yields poor results.

7 CONCLUSION

We presented TopK-InfluMax, a novel algorithmic framework for the top-K influence maximization problem, and InfluNet, a neural network formulation to predict the top-K nodes by infection probability given a seed set of nodes $S \subset V$ as input. We show that TopK-InfluMax beats the state-of-the-art algorithms in the IM space, due to our re-formulation of the utility function as a γ-weakly submodular function minus a modular cost function. We also show that InfluNet out-competes relevant machine learning baselines in the IM space for the top-K IM problem.

A PROOF OF TECHNICAL RESULTS

A.1 Proof of Proposition 2

Figure 5: Counterexample used in the proof of Proposition 2.

Proof by counterexample. Consider an IC model, the graph G and the edge probabilities $\{p_{uv}\}$ in Figure 5. Consider $S = \{1, 2\}$, $T = S \cup \{5\}$ and $u = 3$. Let $K = 1$, and assume that $p_{23} > 0, p_{34} = 0, p_{45} = 1$. Hence, $F_K(S)$ for $K = 1$ would be with the greatest edge-probability in the out-neighborhood of S. Next we compute the values of the objective functions for different seeds.

$$F_K(S) = p_{23} > 0,\ F_K(S \cup \{u\}) = p_{34} = 0, \qquad (16)$$
$$F_K(T) = p_{45} = 1,\ F_K(T \cup \{u\}) = p_{45} = 1$$

We note that $F_K(S \cup \{u\}) - F_K(S) = -p_{23} < 0$ and therefore $F_K(S)$ is not monotone. Moreover, we have $F_K(T \cup \{u\}) - F_K(T) = 0$ which implies, $F_K(S \cup \{u\}) - F_K(S) < F_K(T \cup \{u\}) - F_K(T)$. Therefore, $F_K(S)$ is not submodular.

A.2 Proof of Theorem 3

Proof. Assume a configuration where if $u \in S$, then $q_u = \rho$. In such a case, since $\rho > q_{\max}$ and $p(u; S) = 1$ for $u \in S$, the top-K+$|S|$ in terms of the weighted probabilities will always include S, since the

value of ρ will be always higher that $q_u p_u$. Thus, we define a seed dependent weight $\mu(u;S)$ as follows:

$$\mu(u;S) = q_u \text{ if } u \notin S, \text{and, } \rho \text{ if } u \in S \tag{17}$$

Note that $\mu(u;S)$ is monotone in S. Then we can write:

$$H_K(S) = \max_{I\subseteq V\setminus S:|I|=K} \sum_{u\in I} q_u p(u;S) + \rho|S| \tag{18}$$

$$\overset{(i)}{=} \max_{I\subseteq V\setminus S:|I|\leq K} \sum_{u\in I} q_u p(u;S) + \rho|S| \tag{19}$$

$$\overset{(ii)}{=} \max_{I\subseteq V:|I|\leq|S|+K} \sum_{u\in I} \mu(u;S)p(u;S) \tag{20}$$

Inequality (i) is due to the fact that $\sum_{u\in I}\mu(u;S)p(u;S)$ increases with augmenting more elements I and thus the maximal I will have cardinality K. Inequality (ii) is due to the fact that $q_u p(u;S) > q_v p(v;S)$ for $u \in S$ and $v \notin S$ (since $u \in S$ implies that $q_u = \rho > q_{\max}$). Next we define $I^S = \text{argmax}_{I\subseteq V:|I|\leq|S|+K} \sum_{u\in I}\mu(u;S)p(u;S)$.

$$\begin{aligned}
&H_K(S\cup\{v\}) - H_K(S)\\
&= \sum_{u\in I^{S\cup\{v\}}} \mu(u;S\cup\{v\})p(u;S\cup\{v\}) - \sum_{u\in I^S}\mu(u;S)p(u;S)\\
&\overset{(i)}{\geq} \sum_{u\in I^{S\cup\{v\}}} \mu(u;S\cup\{v\})p(u;S\cup\{v\}) - \sum_{u\in I^S}\mu(u;S)p(u;S\cup\{v\})\\
&\overset{(ii)}{\geq} \sum_{u\in I^{S\cup\{v\}}} \mu(u;S\cup\{v\})p(u;S\cup\{v\})\\
&\qquad - \max_{\substack{I'\subseteq V:\\ |I'|\leq|S|+K}} \sum_{u\in I'} \mu(u;S\cup\{v\})p(u;S\cup\{v\}) \qquad (21)\\
&= \max_{\substack{I\subseteq V:\\ |I|\leq|S|+K+1}} \sum_{u\in I'} \mu(u;S\cup\{v\})\, p(u;S\cup\{v\})\\
&\qquad - \max_{\substack{I'\subseteq V:\\ |I'|\leq|S|+K}} \sum_{u\in I'} \mu(u;S\cup\{v\})\, p(u;S\cup\{v\})\\
&\overset{(iii)}{\geq} q_{\min}\cdot p_{\min} > 0
\end{aligned}$$

Inequality (i) is due to the fact that $p(u;S\cup\{v\}) \geq p(u;S)$ as increasing seed set size will never reduce the infection probability of any node. Inequality (ii) is due to $\mu(u;S\cup\{v\}) > \mu(u;S)$ and the fact that the second term takes the maximum value for all $I' : |I'| \leq |S| + K$. If the solution of the optimization problem of the second term in Eq. (21) is I', then $I^{S\cap\{v\}}\setminus I'$ consists of only one node, say w. Thus, the value of the quantity in Eq. (21) would be $q_w p(w;S)$. Inequality (iii) follows from this.

A.3 Proof for Theorem 4

Proof. We first attempt to bound $H_K(T\cup\{v\}) - H_K(T)$.

$$\begin{aligned}
&H_K(T\cup\{v\}) - H_K(T)\\
&= \sum_{u\in I^{T\cup\{v\}}} \mu(u;T\cup\{v\})p(u;T\cup\{v\}) - \sum_{u\in I^T}\mu(u;T\cup\{v\})p(u;T\cup\{v\})\\
&\qquad + \underbrace{\sum_{u\in I^T}[\mu(u;T\cup\{v\}) - \mu(u;T)]p(u;T\cup\{v\})}_{\leq\rho - q_{\min}}\\
&\qquad + \sum_{u\in I^T}\mu(u;T)p(u;T\cup\{v\}) - \sum_{u\in I^T}\mu(u;T)p(u;T)\\
&\overset{(i)}{\leq} \sum_{u\in I^{T\cup\{v\}}\setminus I^T}\mu(u;T\cup\{v\})p(u;T\cup\{v\}) + \rho - q_{\min}\\
&+ \sum_{u\in I^T\setminus I^S}\mu(u;T)p(u;\{v\}) + \sum_{u\in I^S\cap I^T}\mu(u;T)[p(u;T\cup\{v\}) - p(u;T)]\\
&\overset{(ii)}{\leq} 2\,(|V|-K)\,\rho p_{\max} + \sum_{u\in I^S\cap I^T}\mu(u;T)[p(u;T\cup\{v\}) - p(u;T)]\\
&\quad + \rho - q_{\min} \qquad (22)
\end{aligned}$$

Inequality (i) follows from

$$\begin{aligned}
&\sum_{u\in I^T}\mu(u;T)p(u;T\cup\{v\}) - \sum_{u\in I^T}\mu(u;T)p(u;T)\\
&= \sum_{u\in I^T\setminus I^S}\mu(u;T)[p(u;T\cup\{v\}) - p(u;T)]\\
&\quad + \sum_{u\in I^T\cap I^S}\mu(u;T)[p(u;T\cup\{v\}) - p(u;T)] \qquad (23)
\end{aligned}$$

and the submodularity of $p(u;T)$ in T. Inequality (ii) is due to the fact that $|I^A\setminus I^B| = |I^A\cup I^B| - |I^B| \leq |V| - K$. Next, we reduce the second term in Eq. (22) to the following:

$$\begin{aligned}
&\sum_{u\in I^S\cap I^T}\mu(u;T)[p(u;T\cup\{v\}) - p(u;T)]\\
&\overset{(i)}{\leq} \sum_{u\in I^S}\mu(u;S)\frac{\mu(u;T)}{\mu(u;S)}[p(u;S\cup\{v\}) - p(u;S)]\\
&\overset{(ii)}{\leq} \frac{\rho}{q_{\min}}\left[\sum_{u\in I^{S\cup\{v\}}}\mu(u;S\cup\{v\})p(u;S\cup\{v\}) - \sum_{u\in I^S}\mu(u;S)p(u;S)\right]\\
&= \frac{\rho}{q_{\min}}[H_K(S\cup\{v\}) - H_K(S)] \qquad (24)
\end{aligned}$$

Inequality (i) is due to submodularity of $p(u;S)$. Inequality (ii) is due to $\mu(u;T)/\mu(u;S) \leq \rho/q_{\min}$, $\mu(u;S)$ is monotone in S and $I^{S\cup\{v\}}$ is $\text{argmax}_{I:|I|\leq K+|S|+1}\sum_{u\in I}\mu(u;S\cup\{v\})p(u;S\cup\{v\})$ which provides higher value than the set I^S, of cardinality $|S|+K+1$. Combining Eq. (22) and Theorem 3, we have:

$$\frac{H_K(u\,|\,S)}{H_K(u\,|\,T)} \geq \frac{q_{\min}p_{\min}}{\rho(p_{\min}+1) - q_{\min} + 2(|V|-K)\rho p_{\max}} \tag{25}$$

This gives us an ratio for γ-weak submodularity. Next, we have:

$$H_K(u\,|\,T) = F_K(u\,|\,T) + \rho \leq Kp_{\max} + \rho \tag{26}$$

This proves the second upper bound, *i.e.*, $\gamma^* = q_{\min}p_{\min}/(\rho + Kp_{\max})$

A.4 Proof of Proposition 6

We prove the result by induction. Note that for $n = 0$, $p^{(0)}(v\,|\,S) = \sum_{u\in S}\mathbf{1}[v = u]$ is monotone modular where $\mathbf{1}[\cdot]$ is an indicator function. Now let us assume that $p^{(n)}(v\,|\,S)$ is monotone submodular. Then using the fact that— $\phi(f(S))$ is monotone and submodular function if ϕ is an increasing concave function and f is submodular— we can prove that $p^{(n+1)}(v\,|\,S)$ is also submodular.

REFERENCES

[1] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the Myths of Influence Maximization: An In-Depth Benchmarking Study. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 651–666. https://doi.org/10.1145/3035918.3035924

[2] Cigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V. S. Lakshmanan. 2015. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *PVLDB* 8 (2015), 814–825.

[3] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *SODA*. 946–957.

[4] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to Boost Content Spread in Social Networks. In *WWW*. 529–538.

[5] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *Proceedings of the 21st international conference on World Wide Web*. 529–538.

[6] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*. 1029–1038.

[7] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*. 88–97.

[8] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. 2013. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *CIKM*. 509–518.

[9] Thang N. Dinh, Huiyuan Zhang, Dzung T. Nguyen, and My T. Thai. 2014. Cost-effective Viral Marketing for Time-critical Campaigns in Large-scale Social Networks. *IEEE/ACM Trans. Netw.* 22 (2014), 2001–2011.

[10] Brian W Dolhansky and Jeff A Bilmes. 2016. Deep submodular functions: Definitions and learning. *Advances in Neural Information Processing Systems* 29 (2016).

[11] Pedro Domingos and Matt Richardson. 2001. Mining the Network Value of Customers. In *KDD*. 57–66.

[12] Satoru Fujishige. 2005. *Submodular functions and optimization*. Elsevier.

[13] Satoru Fujishige and Satoru Iwata. 1999. Minimizing a submodular function arising from a concave function. *Discrete applied mathematics* 92, 2-3 (1999), 211–215.

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[15] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. 2010. Learning Influence Probabilities in Social Networks. In *WSDM*. 241–250.

[16] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. 2011. CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In *WWW (Companion Volume)*. 47–48.

[17] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*. 211–220.

[18] M. Granovetter. 1978. Threshold Models of Collective Behavior. *The American Journal of Sociology* 83, 6 (1978), 1420–1443.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[20] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. 2019. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In *International Conference on Machine Learning*. PMLR, 2634–2643.

[21] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. 2012. Influence Blocking Maximization in Social Networks under the Competitive Linear Threshold Model. In *SDM*. 463–474.

[22] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and Robust Influence Maximization in Social Networks. In *ICDM*. 918–923.

[23] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence Through a Social Network. In *KDD*. 137–146.

[24] Arijit Khan, Benjamin Zehnder, and Donald Kossmann. 2016. Revenue maximization by viral marketing: A social network host's perspective. In *ICDE*. 37–48.

[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[26] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[27] Konstantin Kutzkov, Albert Bifet, Francesco Bonchi, and Aristides Gionis. 2013. STRIP: Stream Learning of Influence Probabilities. In *KDD*. 275–283.

[28] W. Lee, J. Kim, and H. Yu. 2012. CT-IC: Continuously Activated and Time-Restricted Independent Cascade Model for Viral Marketing. In *ICDM*. 960–965.

[29] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective Outbreak Detection in Networks. In *KDD*. 420–429.

[30] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[31] Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*. 510–520.

[32] B. Liu, G. Cong, D. Xu, and Y. Zeng. 2012. Time Constrained Influence Maximization in Social Networks. In *ICDM*. 439–448.

[33] Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V. S. Lakshmanan. 2013. The bang for the buck: fair competitive viral marketing from the host perspective. In *KDD*. 928–936.

[34] Sahil Manchanda, Akash Mittal, Anuj Dhawan, Sourav Medya, Sayan Ranu, and Ambuj Singh. 2020. Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs. *Advances in Neural Information Processing Systems* 33 (2020), 20000–20011.

[35] Christopher D Manning. 2008. *Introduction to information retrieval*. Syngress Publishing,.

[36] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-Stare: Optimal Sampling Algorithms for Viral Marketing in Billion-scale Networks. In *SIGMOD*. 695–710.

[37] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations. In *AAAI*. 138–144.

[38] George Panagopoulos, Fragkiskos Malliaros, and Michalis Vazirgiannis. 2020. Multi-task learning for influence estimation and maximization. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[39] Matthew Richardson and Pedro Domingos. 2002. Mining Knowledge-sharing Sites for Viral Marketing. In *KDD*. 61–70.

[40] Thomas C. Schelling. 1978. *Micromotives and Macrobehavior*.

[41] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 240–248.

[42] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.

[43] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD*. 1539–1554.

[44] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence Maximization: Near-optimal Time Complexity Meets Practical Efficiency. In *SIGMOD*. 75–86.

[45] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *stat* 1050 (2017), 20.