Influence Maximization with Tag Revisited: Exploiting the Bi-submodularity of the Tag-Based Influence Function

Atharva Tekawade and Suman Banerjee(✉)

Department of Computer Science and Engineering, Indian Institute of Technology Jammu, Jammu 181221, India
{2018uee0137,suman.banerjee}@iitjammu.ac.in

Abstract. Given a Social Network how to select a small number of influential users to maximize the influence in the network has been studied extensively in the past two decades and formally referred to as the Influence Maximization Problem. Among most of the existing studies, it has been implicitly assumed that there exists a single probability value that represents the influence probability between the users. However, in reality, the influence probability between any two users is dependent on the context (formally referred to as tag e.g.; a sportsman can influence his friends related to any news related to sports with high probability). In this paper, we bridge the gap by studying the Tag-Based Influence Maximization Problem. In this problem, we are given with a social network where each edge is marked with one probability value for every tag and the goal here is to select k influential users and r influential tags to maximize the influence in the network. First, we define a tag-based influence function and show that this function is bi-submodular. We use the orthent-wise maximization procedure of bi-submodular function which gives a constant factor approximation guarantee. Subsequently, we propose a number of efficient pruning techniques that reduces the computational time significantly. We perform an extensive number of experiments with real-world datasets to show the effectiveness and efficiency of the proposed solution approaches.

Keywords: Social Network · Influence Maximization · Seed Set · Bi-Submodular Function

1 Introduction

Diffusion of information in a networked system has been studied extensively to answer several questions in different domains such as how infectious disease spreads in a human contact network [9], how malware, wormholes, etc. spread in computer networks [1], how innovation, concepts, ideas, etc. spread through

The work of Dr. Suman Banerjee is supported with the Seed Grant sponsored by the Indian Institute of Technology Jammu (Grant No.: SG100047).

X. Yang et al. (Eds.): ADMA 2023, LNAI 14176, pp. 772–786, 2023.
https://doi.org/10.1007/978-3-031-46661-8_51

social networks and many more [6]. The diffusion of information in social networks has got applications in different domains such as *viral marketing* [13], *computational advertisement* [10], *feed ranking*, etc. Hence, several diffusion models have been proposed in the literature. Among them, one of the popular diffusion models that have been studied extensively is the *Independent Cascade Model* (abbreviated as *IC Model*). According to this model, information is diffused in discrete time steps from a set of initially active nodes called as *seed nodes*. A node can be any one of the following two states: '*active*' (also called as '*influenced*') and '*inactive*' (also called as '*influenced*'). Every active node at time step t will get a single chance to activate its inactive neighbors with the success probability as the edge weight. The diffusion process ends when no more node activation is possible.

One important problem that has been studied in the context of information diffusion is the problem of *Influence Maximization*. Given a social network and a positive integer k the problem of influence maximization asks to choose a subset of k nodes whose initial activation leads to maximum influence in the network. This problem has got a significant applications in the domain of viral marketing. Consider a commercial house developed a new product and wants to prompt among people. They distribute a limited number of sample product (in free of cost or in discounted price) among a group of highly influential users with a hope that they will use the product and share information about it among their neighbors. Some of them will be influenced and buy the product. This cascading process will go on and at the end of diffusion process a significant number of people will ultimately buy this product and the E-Commerce house can earn revenue. Due to practical applications the problem of influence maximization has been studied extensively and several solution methodologies has been proposed in the literature.

One of the important drawback of the existing studies is that most of them considered that there exists a single influence value between any two users of the network. However, in practice the case is not exactly the same. The influence probability between any two user is always dependent on the context. As an example, a sportsman can influence related to any issue related to sports with high probability compared to any other contexts. These contexts are formalized as tags. In real-world situations between every pair of users there exists an influence probability value corresponding to every tag. Now it is easy to observe that the influence in the network will not only depends on the seed set we are choosing, but also the tags we are choosing. In this context the problem that arises is given a social network where each edges of the network is marked with an influence probability value corresponding to every tag and the aim is to choose a subset of k nodes and r tags such that the influence in the network gets maximized. Though this problem is quite natural in many realistic situations, however we observe that the number of studies which considers both selection of tags and seed nodes for influence maximization is very limited.

To the best of our knowledge, Ke et al. [7] were the first to study this problem and they proposed a sketch-based solution approach for this problem. Also,

their experiments show that their proposed methodology can process significantly large datasets within reasonable computational time. Banerjee et al. [3,4] studied a similar problem where they considered the users and tags have non-uniform selection costs and a fixed amount of budget is given. The goal was to select a subset of the nodes as seed nodes and a subset of the tags within the budget to maximize the influence. They showed that in many keyword-based Social Network datasets the popularity of the tags varies a lot across different communities within the same network. Other than these two studies there are no studies that considers the same problem. In both these studies there is no mathematical analysis of the tag-based influence function has been done. One key observation of this study is that this analysis leads to efficient algorithms for optimization of this function. In particular, we make the following contributions in this paper:

- We study the problem of selecting influential users and tags simultaneously for which there exists limited studies in the literature.
- We do a mathematical analysis of the Tag-Based Influence Function and prove several theoretical results.
- We propose a Coordinate-wise Solution Approach and Community-based Solution Approach to solve the Tag-Based Influence Maximization Problem.
- We perform an extensive set of experiments with real-life datasets to show the effectiveness and efficiency of the proposed solution approaches.

Rest of the paper is organized as follows. Section 2 describes relevant preliminary concepts and defines our problem formally. Section 3 describes the proposed solution approaches with detailed analysis. Section 4 contains the experimental validation of the proposed solution approaches and finally, Sect. 5 concludes this study and gives future research directions.

2 Preliminaries and Problem Definition

In this section we describe some preliminary concepts and defines our problem formally. Initially we start by describing social networks.

2.1 Social Network

In this study we model a social network by a weighted and directed graph $G(V, E, P)$ where the vertex set $V(G) = \{u_1, u_2, \ldots, u_n\}$ represents the set of users of the network. The edge set $E(G)$ are the set of social ties among the users; i.e., there is an edge between the users u_i and u_j if there exists a social relation between u_i and u_j. We denote the number of vertices and edges of G by n and m, respectively. Consider there are a set of tags $\mathbb{T} = \{t_1, t_2, \ldots, t_k\}$ which are relevant to the users. For every edge $(u_i u_j) \in E(G)$ and for every tag $t \in \mathbb{T}$ there exists an influence probability denoted by $\mathcal{P}^t_{u_i \to u_j}$. This can be interpreted as as the influence probability of the edge (uv) when the tag t is used for the

diffusion process. In the graph G, the edge weight function $\mathcal{P}$ maps each edge-tag pair to the corresponding influence probability; i.e.; $\mathcal{P} : E(G) \times \mathbb{T} \longrightarrow (0, 1]$. Now, we can observe that for all the edges their tag specific probability can be represented by a $m \times k$ matrix denoted by $\mathbb{P}$. (i, j)-th entry of the matrix $\mathbb{P}$; i.e.; $\mathbb{P}[i, j]$ contains the influence probability of the edge e_i for the tag t_j. Now, for a given subset of tags $T^{'} \subseteq \mathbb{T}$, how to aggregate the influence probabilities for the tags in $T^{'}$ to obtain the influence probability of the edge. This depends on how we are aggregating the tags. In this study we are aggregating the tags considering they are independent to each other and this called as independent tag aggregation which is stated in Definition 1.

Definition 1 (Independent Tag Aggregation). *For an edge $(u_i u_j) \in E(G)$ and a subset of the tags $T^{'} \subseteq \mathbb{T}$ the aggregated influence probability of this edge is denoted as $\mathcal{P}^{T^{'}}_{u_i \rightarrow u_j}$ and defined using Equation No. 1.*

$$\mathcal{P}^{T^{'}}_{u_i \rightarrow u_j} = 1 - \prod_{t \in T^{'}} (1 - \mathcal{P}^{t}_{u_i \rightarrow u_j}) \tag{1}$$

Now, it is easy to observe that for all the edges in the worst case the independent tag aggregation can take $\mathcal{O}(k \cdot m)$ time.

2.2 Influence Diffusion in Social Networks

The diffusion process in a networked system has been studied extensively due to its applications in different domains including Epidemiology, Computer Networks, Social Networks, and many more. As this paper deals with social networks, here we discuss the diffusion of information in social networks. Due to several application domains such as viral marketing, computational advertisement, feed ranking, etc. there are extensive studies on the diffusion of information in social networks. Several models have been proposed and studied in the literature. Among them, two fundamental models that have been considered extensively are the *Independent Cascade Model (IC Model)* and *Linear Threshold Model (LT Model)*. In this study, we consider the diffusion in the underlying network is happening according to the rule of the IC Model which is stated in Definition 2.

Definition 2 (Independent Cascade Model). *The rules of the independent cascade model is as follows:*

- *Information is diffused in discrete time steps.*
- *A node can be either of the two states:* 'inactive' *(also refereed to as* 'uninfluenced*') and* active *(also referred to as* 'influenced*')*
- *An active node at time step t will get a single chance to activate its inactive neighbors at time step $(t + 1)$.*
- *A node can change its state from active to 'inactive', however not the vice versa.*

Let, $T^{'}$ be the set of tags that are used for the diffusion process. Also, these tags are aggregated as per Equation No. 6. In IC Model information is diffused in discrete time steps. It is assumed that initially (i.e.; at time step $t = 0$) a subset of the nodes $\mathcal{S} \subseteq V(G)$ are active and the diffusion process starts from the nodes in $\mathcal{S}$. We call these nodes as *Seed Nodes*. Every active node at time step t will get a single chance to activate its inactive neighbors with success probability as the aggregated edge probability. Now in the diffusion process, some of the nodes will be active. $I(\mathcal{S})$ denotes the set of nodes that are activated from the seed set $\mathcal{S}$. The number of nodes in $I(\mathcal{S})$ is called the *Influence of the Seed Set* $\mathcal{S}$. For any seed set $\mathcal{S}$, its influence is denoted as $\sigma(\mathcal{S})$ which is stated in Definition 3.

Definition 3 (Influence of a Seed Set). *Given a seed set $\mathcal{S}$, its influence is denoted by $I(\mathcal{S})$ and defined as the number of nodes that are activated at the end of diffusion process. Hence, $\sigma(\mathcal{S}) = |I(\mathcal{S})|$. Here, $\sigma()$ is the influence function that maps each subset of the nodes to its expected influence; i.e.; $\sigma : 2^{V(G)} \longrightarrow \mathbb{R}_0^+$ with $\sigma(\emptyset) = 0$.*

Now it is important to observe that in our problem we are dealing with both tags and influential users. Hence, we have to extend the influence function to the Tag-Based Influence Function which is described in the next subsection.

2.3 Tag-Based Social Influence

For any positive integer i, $[i]$ denotes the set $\{1, 2, \ldots, i\}$. As mentioned in Definition 3 given a seed set $\mathcal{S}$, the social influence function $\sigma(.)$ returns its influence. However, as in this study we are dealing with both users and tag we have to generalize the influence function that takes two arguments one is a subset of nodes and the other one is a subset of the tags. We denote the tag-based influence function as $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ and stated in Definition 4.

Definition 4 (Tag-Based Influence Function). *Given a subset of the nodes $\mathcal{S} \subseteq V(G)$ and a tag set $T^{'} \subseteq \mathbb{T}$ the tag-based influence function returns the influence if the seed set $\mathcal{S}$ and the tag set $T^{'}$ is used. Hence, $\sigma^{\mathcal{T}} : 2^{V(G)} \times 2^{\mathbb{T}} \longrightarrow \mathbb{R}_0^+$.*

It is an important point to observe that for any subset of nodes $\mathcal{S} \subseteq V(G)$ if no tag is selected the influence will be the cardinality of $|\mathcal{S}|$; i.e.; $\sigma^{\mathcal{T}}(\mathcal{S}, \emptyset) = |\mathcal{S}|$. Now, based on the definition of tag-based influence function we define the problem of tag-based influence maximization which is stated in Definition 5.

Definition 5 (Tag-Based Influence Maximization Problem). *Given a social network $G(V, E, P)$, a set of Tags T, and two positive integers k and r the problem of* Tag-Based Influence Maximization Problem *asks to choose k seed nodes and r tags such that the tag-based influence function $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ is maximized. Mathematically, this problem can be presented using Equation No. 2.*

$$\sigma^{\mathcal{T}}(\mathcal{S}^*, T^{'*}) = \underset{\substack{\mathcal{S} \subseteq V(G) \wedge |\mathcal{S}| \leq k \\ and \\ T^{'} \subseteq \mathbb{T} \wedge |T^{'}| \leq r}}{argmax} \sigma^{\mathcal{T}}(\mathcal{S}, T^{'}) \tag{2}$$

Here, $\mathcal{S}^*$ and $T^{'*}$ denotes the optimal k size seed set and r size tag set, respectively.

It has been mentioned in [8] that the problem of influence maximization is NP-hard and hard to approximate beyond a constant factor under the both IC and LT Mdel of diffusion.

2.4 Set Function and Its Properties

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ be a set with n elements. A function is said to be a set function defined on the ground set $\mathcal{X}$ if f maps every subset of $\mathcal{X}$ to a real number. In this paper, we consider that range of f is the set of positive real numbers including 0; $f : 2^{\mathcal{X}} \longrightarrow \mathbb{R}_0^+$. We say that f is *non-negative* if for any $\mathcal{S} \subseteq \mathcal{X}$, $f(\mathcal{S}) \geq 0$, *monotone* if for all $\mathcal{S} \subseteq \mathcal{X}$ and for all $x \in \mathcal{X} \setminus \mathcal{S}$, $f(\mathcal{S} \cup \{x\}) \geq f(\mathcal{S})$; and submodular if for all $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{X}$ and for all $x \in \mathcal{X} \setminus \mathcal{S}_2$, $f(\mathcal{S}_1 \cup \{x\}) - f(\mathcal{S}_1) \geq f(\mathcal{S}_2 \cup \{x\}) - f(\mathcal{S}_2)$. We say that f is normalized if $f(\emptyset) = 0$. Now, f is said to be a bi-set function if no. of arguments of f are 2. For a bi-set function the ground set of the first and second argument may be same or different. A bi-set function is said to be normalized if for the both the arguments when it is $\emptyset$ then the functional value is 0; i.e.; $f(\emptyset, \emptyset) = 0$. It can be observed that the tag-based influence function $\sigma^{\mathcal{T}}(\mathcal{S}^*, T^{'*})$ is a bi-set function and the ground set of the first argument is the set of nodes of G and the ground set of the second argument is the set of tags $\mathbb{T}$. Now we list down several properties of bi-set functions which be used to analyze the properties of the tag-based influence function [2,11]. Now, we state the notion of *Bi-Monotonicity* in Definition 7.

Definition 6 (Bi-Monotonicity). *A bi-set function f where the ground sets for the first and second arguments $\mathcal{X}$ and $\mathcal{Y}$, respectively is said to be bi-submodular if for all $(\mathcal{A}, \mathcal{B}) \in 2^{\mathcal{X}} \times 2^{\mathcal{Y}}$ and for all $x \in \mathcal{X} \setminus \mathcal{A}$ and $y \in \mathcal{Y} \setminus \mathcal{B}$, $f(\mathcal{A} \cup \{x\}, \mathcal{B}) \geq f(\mathcal{A}, \mathcal{B})$ and $f(\mathcal{A}, \mathcal{B} \cup \{y\}) \geq f(\mathcal{A}, \mathcal{B})$.*

Definition 7 (Bi-Submodularity). *A bi-set function f where the ground sets for the first and second arguments $\mathcal{X}$ and $\mathcal{Y}$, respectively is said to be bi-submodular if for all $(\mathcal{A}, \mathcal{B}) \in 2^{\mathcal{X}} \times 2^{\mathcal{Y}}$, $(\mathcal{A}^{'}, \mathcal{B}^{'}) \in 2^{\mathcal{X}} \times 2^{\mathcal{Y}}$ with $\mathcal{A} \subseteq \mathcal{A}^{'}$ and $\mathcal{B} \subseteq \mathcal{B}^{'}$, $x \notin \mathcal{A}^{'}$ and $y \notin \mathcal{B}^{'}$ if the following two conditions holds:*

$$f(\mathcal{A} \cup \{x\}, \mathcal{B}) - f(\mathcal{A}, \mathcal{B}) \geq f(\mathcal{A}^{'} \cup \{x\}, \mathcal{B}^{'}) - f(\mathcal{A}^{'}, \mathcal{B}^{'}) \tag{3}$$

and

$$f(\mathcal{A}, \mathcal{B} \cup \{y\}) - f(\mathcal{A}, \mathcal{B}) \geq f(\mathcal{A}^{'}, \mathcal{B}^{'} \cup \{y\}) - f(\mathcal{A}^{'}, \mathcal{B}^{'}) \tag{4}$$

3 Proposed Solution Approaches

In this section we describe the proposed solution approaches for our problem. Initially, we start by establishing several properties of the tag-based influence function which will be required for designing algorithms for optimizing the tag-based influence function.

3.1 Properties of the Tag-Based Influence Function

Lemma 1. *The tag-based influence function as defined using Equation No. 2 follows bi-monotonicity property.*

Proof. Consider $\mathcal{S} \subseteq V(G)$ and $u \in V(G) \setminus \mathcal{S}$. Also $T^{'} \subseteq \mathbb{T}$ and $t \in \mathbb{T} \setminus T^{'}$. So the tag-based influence function $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ and to show that $\sigma^{\mathcal{T}}(.,.)$ is a bi-monotone function if both he following is true: (i) $\sigma^{\mathcal{T}}(\mathcal{S} \cup \{u\}, T^{'}) \geq \sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$, and (ii) $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'} \cup \{t\}) \geq \sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$. Case (i) can be observed very easily. Consider $\mathcal{S}^{'} = \mathcal{S} \cup \{v\}$. As in both right and left hand side of Case (i) the tag set remains same, hence the aggregated influence probability will also remain the same. Now it is a fact that under the IC Model of diffusion the influence function is monotone; i.e.; $\sigma(\mathcal{S}^{'}) \geq \sigma(\mathcal{S})$. So the Case (i) is proved.

Now, to prove Case (ii), let us have the observation that for any two tag sets $T^{'}$ and $T^{''}$ with $|T^{''}| > |T^{'}|$. Then for every edge $(uv) \in E(G)$, $\mathcal{P}^{T^{''}}_{u \to v} \geq \mathcal{P}^{T^{'}}_{u \to v}$. So, even if the seed set remains the same if more tags are used in the influence maximization then the influence when more tags are used influence will be more. This proves Case (ii) and as a whole the lemma statement.

Lemma 2. *The tag-based influence function as defined using Equation No. 2 follows bi-submodularity property.*

Proof. Consider $\mathcal{S}^{'} \subseteq \mathcal{S}^{''} \subseteq V(G)$ and $u \in V(G) \setminus \mathcal{S}^{''}$. Also, $T^{'} \subseteq T^{''} \subseteq \mathbb{T}$ and $t \in \mathbb{T} \setminus T^{''}$. Now, the tag-based influence function $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ will said to be a bi-submodular set function if both of the following are true:

- **Case I:** First we show that $\sigma^{\mathcal{T}}(\mathcal{S}^{'} \cup \{u\}, T^{'}) - \sigma^{\mathcal{T}}(\mathcal{S}^{'}, T^{'}) \geq \sigma^{\mathcal{T}}(\mathcal{S}^{''} \cup \{u\}, T^{'}) - \sigma^{\mathcal{T}}(\mathcal{S}^{''}, T^{'})$. It is easy to observe that in both the left and right hand side of the inequalities, the tag set remains the same. That means the aggregated influence probability for all the edges remains the same. Hence, this case boils down to the simple influence function; i.e.; $\sigma(\mathcal{S}^{'} \cup \{u\}) - \sigma(\mathcal{S}^{'}) \geq \sigma(\mathcal{S}^{''} \cup \{u\}) - \sigma(\mathcal{S}^{''})$ where $\mathcal{S}^{'} \subseteq \mathcal{S}^{''} \subseteq V(G)$ and $u \in V(G) \setminus \mathcal{S}^{''}$. It has been shown by Kempe et al. [7] that under the Independent Cascade Model the influence function is submodular. Hence, Case I is proved. In other words, the tag-based influence function $\sigma^{\mathcal{T}}(.,.)$ is submodular with respect to the first orthent.
- **Case II.** Now, we want to show that $\sigma^{\mathcal{T}}(\mathcal{S}^{'}, T^{'} \cup \{t\}) - \sigma^{\mathcal{T}}(\mathcal{S}^{'}, T^{'}) \geq \sigma^{\mathcal{T}}(\mathcal{S}^{'}, T^{''} \cup \{t\}) - \sigma^{\mathcal{T}}(\mathcal{S}^{'}, T^{''})$. In this case, we can observe that in both sides of the inequalities, the seed set remains the same only the tag set is changing. Now as mentioned previously, for any two tag sets $T^{'}$ and $T^{''}$ such that $|T^{''}| > |T^{'}|$ for any edge $(u, v) \in E(G)$, $\mathcal{P}^{T^{''}}_{u \to v} > \mathcal{P}^{T^{'}}_{u \to v}$. Now consider the standard influence maximization problem in two different cases. In both cases, the topology and the structure of the graph is the same, however the edge probabilities are different. Let, $G^{T^{'}}$ and $G^{T^{''}}$ are the input social network with the aggregated edge probabilities for the tags $T^{'}$ and $T^{''}$, respectively. Let, $\sigma_{G^{T^{'}}}(\mathcal{S})$ and $\sigma_{G^{T^{''}}}(\mathcal{S})$ denote the influence of the seed set $\mathcal{S}$ on the graphs $G^{T^{'}}$ and $G^{T^{''}}$, respectively. Also, it is easy to observe that $\sigma_{G^{T^{''}}}(\mathcal{S}) > \sigma_{G^{T^{'}}}(\mathcal{S})$.

This proves that the tag-based influence function $\sigma^{\mathcal{T}}(.,.)$ is bi-submodular.

The Bi-Submodularity property as mentioned in Lemma 2 has been exploited in the proposed solution methodology as described in the following sub-section.

3.2 Proposed Solution Approach

In this section we describe the proposed solution methodology which is based on the orthent-wise maximization of a bi-submodular function as described below.

Broad Idea of the Proposed Solution Approach. As mentioned in Sect. 2.3, the problem here is to maximize the bi-submoular set function $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ subject to the constraint $|\mathcal{S}| \leq k$ and $|T^{'}| \leq r$. Now, if we apply the coordinate wise maximization algorithm that works in the following way:

- **Step 1:** First, the tag set is initialized to an empty set and find out an optimal k-sized seed set that maximizes the tag-based influence function. Consider the obtained seed set is $\mathcal{S}^*$. So, we are solving the following optimization problem:

$$\mathcal{S}^* \longleftarrow \underset{S \subseteq V(G) \text{ and } |\mathcal{S}| \leq k}{argmax} \sigma^{\mathcal{T}}(\mathcal{S}, \emptyset) \tag{5}$$

- **Step 2:** Once the optimal seed set is found, we fix the seed set in the tag-based influence function with the optimal seed set, and we find out an optimal r-size tag set and let it be T^*. So, in this step, we are solving the following optimization problem:

$$T^* \longleftarrow \underset{T' \subseteq \mathbb{T} \text{ and } |T'| \leq r}{argmax} \sigma^{\mathcal{T}}(\mathcal{S}^*, T^{'}) \tag{6}$$

Intuitively, we can observe that after solving the optimization problems mentioned in Equations No. 5 and 6 we will obtain both the k-size seed set and r-size tag set. However, this is not possible if we apply both steps directly. The reason behind this is as follows. Consider the case of solving the optimization problem mentioned in Equation No. 5. When we assign the tag set to an empty set, the aggregated influence probability for all the edges of the network will be 0, which is equivalent to a graph with n nodes but no edges at all. Now, on such network we try to find an optimal k-sized seed set for the influence maximization problem using the incremental greedy approach based on marginal influence gain then any k-size subset of the vertex set can be returned which is not correct. So, in the proposed solution approach we tackle this problem and describe the proposed solution approach.

Description of the Proposed Solution Approach. We tackle the above mentioned problem in the following way. Initially, we choose the most popular tag and subsequently we select $(r-1)$ many tags incrementally after selecting the k-size seed set. So the working principle of the proposed solution approach is as follows. First we initialize the seed set and tag set to empty set. Then we select

the most popular tag in the network and this can be done in the following way. Every user of the network is marked with the tags that they are associated with. Now, for every tag we count the frequency of every tag and choose the highest one. If there are ties that can be broken arbitrarily. This highest frequency tag is selected, and subsequently, we are left with to select k seed nodes and $(r-1)$-tags.

Algorithm 1: Co-Ordinate wise Maximization Algorithm for the Tag-Based Influence Maximization Problem

Data: The Social Network $G(V, .E, \mathcal{P})$, the tag set $\mathbb{T}$, Two positive integer k and r.
Result: $\mathcal{S} \subseteq V(G)$ with $|\mathcal{S}| = r$ and $T \subseteq \mathbb{T}$ with $|T| = r$ such that $\sigma^{\mathcal{T}}(\mathcal{S}, T)$ is maximized

1 $\mathcal{S} \longleftarrow \emptyset$; $T \longleftarrow \emptyset$;
2 $t \longleftarrow$ The most popular tag; $T \longleftarrow T \cup \{t\}$;
3 **for** $i = 1$ *to* k **do**
4 $\quad u^* \longleftarrow \underset{u \in V(G) \setminus \mathcal{S}}{argmax}\ \sigma^{\mathcal{T}}(\mathcal{S} \cup \{u\}, T) - \sigma^{\mathcal{T}}(\mathcal{S}, T)$; $\mathcal{S} \longleftarrow \mathcal{S} \cup \{u^*\}$;
5 **end**
6 **for** $j = 1$ *to* $(r-1)$ **do**
7 $\quad t^* \longleftarrow \underset{t' \in \mathbb{T} \setminus T}{argmax}\ \sigma^{\mathcal{T}}(\mathcal{S}, T \cup \{t'\}) - \sigma^{\mathcal{T}}(\mathcal{S}, T)$; $T \longleftarrow T \cup \{t^*\}$;
8 **end**
9 *return* $\mathcal{S}, T$;

Algorithm 1 describes the proposed solution approach in terms of pseudocode. In all the social network datasets where tags are associated every user of the network will be marked with the tags that they are associated with. From the frequency of tags the most frequent tag can be identified and taken into the tag set T. Consider the number of tags in the dataset is p. Now, finding the highest frequency tag can be obtained in $\mathcal{O}(p \cdot n)$ time. Line no. 6 computes the marginal gain and in the worst case in each iteration of the `for` loop of Line No. 5, the number of marginal gains computed is of $\mathcal{O}(n)$. Now, to compute the influence while computing the influence of a seed set under the independent cascade model of diffusion is of $\mathcal{O}(n \cdot (m+n))$. The `for` loop of Line No. 6 will execute for $\mathcal{O}(k)$ times. Hence, the time requirement to execute from Line No. 5 to 7 is of $\mathcal{O}(k \cdot n^2 \cdot (m+n))$. There is a little difference with the execution of the `for` loop of Line No. 8 because in each iteration the newly selected tag has to be aggregated for computing the marginal gain in the next iteration. As the number of tags is of $\mathcal{O}(p)$, then for all the edges to aggregate the influence probabilities will take $\mathcal{O}(p \cdot m)$ time. This additional time we have to bear in each iteration. Hence, the time requirement for executing from Line No. 8 to 10 will be $\mathcal{O}(r \cdot (pm + n^2 \cdot (m+n)))$. Hence, the total time requirement by Algorithm 1 is of $\mathcal{O}(p \cdot n + k \cdot n^2 \cdot (m+n) + r \cdot (pm + n^2 \cdot (m+n)))$. Now, the space requirement will be as follows will be of $\mathcal{O}(n)$ in the worst case to store $\mathcal{S}$, $\mathcal{O}(r)$ in the worst case to store T, $\mathcal{O}(m)$ to store the aggregated influence probabilities, also $\mathcal{O}(n)$ and $\mathcal{O}(r)$ to store the marginal influence gains while executing the `for` loop of

Line No. 5 and 8, respectively. Hence, the total space requirement will be of $\mathcal{O}(m+r)$. So, Theorem 1 holds.

Theorem 1. *Time and space requirement of Algorithm 1 will be of* $\mathcal{O}(p \cdot n + k \cdot n^2 \cdot (m+n) + r \cdot (pm + n^2 \cdot (m+n)))$ *and* $\mathcal{O}(m+r)$*, respectively.*

Community-Based Approach. In this section, we propose a community-based approach for solving the tag-based influence maximization problem. In this approach, the input social network is divided among the communities. The budget for both seed node and tags are divided among the communities based on the following criteria: "If the size of the community is large then it requires more seed nodes and tags to influence". Let, the network is divided into ℓ many communities and they are represented by $C = \{C_1, C_2, \ldots, C_\ell\}$. Let, for any community $C_i \in C$, $V(C_i)$ and $E(C_i)$ denote the set of vertices and edges of the community C_i. Also for any two communities $C_i, C_j \in C$, E_{C_i,C_j} denotes the set of edges between C_i and C_j; i.e.; $E_{C_i,C_j} = \{(uv) : u \in V(C_i) \text{ and } v \in V(C_j)\}$. So, $V(G) = \bigcup_{C_i \in C} V(C_i)$ and $E(G) = \bigcup_{\substack{C_i,C_j \in C \\ \text{and } C_i \neq C_j}} (E_{C_i,C_j} \cup E_{C_i})$. For any community $C_i \in C$, we denote the number of nodes and edges of this community are denoted by n_i and m_i, respectively. So, we have the following $n = \sum_{i \in [\ell]} n_i$. Also, for any pair of communities C_i and C_j, let m_{ij} denotes the number of edges between the community C_i and C_j. If the graph is undirected then m_{ij} and m_{ji}. Now, we divide the budget for nodes and tags among the communities as follows:

- **Budget Division for Seed Nodes** For any Community C_i, the number of maximum seed nodes that can be selected from this community can be given by $(\frac{k}{n} \cdot n_i)$.
- **Budget Division for Tags** For any Community C_i, the maximum number of tags that can be selected from this community can be given by $(\frac{r-1}{m} \cdot m_i)$.

It is important to observe that after the division of the budgets among the communities, the total budgets for all the communities do not exceed the allocated budget. Once the budget division is done the next step is to choose the seed nodes and tags using any algorithm. In this approach, we use the high-frequency tags in the community within the budget and we select the seed nodes based on the marginal influence gain. The proposed methodology has been described in terms of psudocode in Algorithm 2.

Now we describe the working principle of Algorithm 2. First, we initialize two sets $\mathcal{S}$ and T to store the seed nodes and tags, respectively. Next, we detect the communities of the network and for this purpose, we use the Louvian algorithm [12]. Here, *Community* is an array of size n where n is the number of nodes of G. Its i-th contains the community number to which the vertex v_i belongs to. So, `Community[i]` $= x$ means the vertex v_i belongs to the x-th community. Also, it is easy to observe that the maximum value among the numbers of this list gives the number of communities in which the network has been divided. Next,

Algorithm 2: Community-Based Approach for the Tag-Based Influence Maximization Problem

Data: The Social Network $G(V, .E, \mathcal{P})$, the tag set $\mathbb{T}$, Two positive integer k and r.
Result: $\mathcal{S} \subseteq V(G)$ with $|\mathcal{S}| = r$ and $T \subseteq \mathbb{T}$ with $|T| = r$ such that $\sigma^{\mathcal{T}}(\mathcal{S}, T)$ is maximized

1 $\mathcal{S} \longleftarrow \emptyset$; $T \longleftarrow \emptyset$;
2 $Community = Community\_Detection(G)$; $\ell \longleftarrow max(Community)$;
3 $Seed\_Budget \longleftarrow array(\ell, 0)$; $Tag\_Budget \longleftarrow array(\ell, 0)$;
4 $Community\_Size \longleftarrow array(\ell, 0)$;
5 **for** $i = 1$ *to* n **do**
6 | **if** $Community[i] == x$ **then**
7 | | $Community\_Size[x] = Community\_Size[x] + 1$;
8 | **end**
9 **end**
10 **for** $i = 1$ *to* ℓ **do**
11 | $Seed\_Budget[i] \longleftarrow \frac{n_i}{n} \cdot k$; $Tag\_Budget[i] \longleftarrow \frac{m_i}{m} \cdot r$;
12 **end**
13 **for** $i = 1$ *to* ℓ **do**
14 | $T_i \longleftarrow \emptyset$;
15 | $Tag\_Popularity \longleftarrow$ Calculate the Tag Popularity of the i-th Community ;
16 | $Sorted\_Tags \longleftarrow$ Sort the tags based on the $Tag\_Popularity$ value;
17 | **for** $j = 1$ *to* $|Sorted\_Tags|$ **do**
18 | | **if** $Sorted\_Tags[j] \notin T$ *and* $|T_i| < Tag\_Budget[i]$ **then**
19 | | | $T_i \longleftarrow T_i \cup \{Sorted\_Tags[j]\}$;
20 | | **end**
21 | **end**
22 | $T \longleftarrow T \cup T_i$;
23 **end**
24 **for** $i = 1$ *to* m **do**
25 | $\mathcal{P}_i^T \longleftarrow$ Calculate the aggregated influence probability ;
26 **end**
27 **for** $i = 1$ *to* ℓ **do**
28 | **for** $j = 1$ *to* $Seed\_Budget[i]$ **do**
29 | | $u^* \longleftarrow \underset{u \in V(G) \setminus \mathcal{S}}{argmax}\ \sigma^{\mathcal{T}}(\mathcal{S} \cup \{u\}, T) - \sigma^{\mathcal{T}}(\mathcal{S}, T)$; $\mathcal{S} \longleftarrow \mathcal{S} \cup \{u^*\}$;
30 | **end**
31 **end**
32 *return* $\mathcal{S}, T$;

we initialize two arrays $Seed\_Budget$ and Tag_Budget of size ℓ, and the i-th entry of both store the budget for seed nodes and tags for the i-th community, respectively. As per the budget division policy described previously, the budget for both seed nodes and tags are divided among the communities from Line No. 6 to 8. The array $Community\_Size$ stores the size of each community; i.e.; x-th entry stores the number of nodes of the x-th community. Now, we subsequently proceed with the tag selection process in the following way. First, we initialize

the set which will store the tag set selected from that community. Now, for each tag, we calculate the frequency of each tag, and subsequently, we sort the tags based on the tag popularity value. From this sorted tag list we scan over this list and while scanning we check whether the current tag has already been selected or not. If not and the allocated budget for that community still has not been exhausted then this tag is chosen. Once the tag selection of a community has been done then they are merged with the global tag set T. Once the tag selection is completed next the tag aggregation is done as mentioned in Equation No. 1. At last, from each of the communities we select the required number of tags based on the marginal influence gain. Next, we analyze the time and space requirement of Algorithm 2.

Initializing both $\mathcal{S}$ and T requires $\mathcal{O}(1)$ time. As mentioned in [5], the time requirement by Louvain Method for detecting communities of the network requires $\mathcal{O}(n \log n)$ time where n denotes the number of nodes of the network. Initializing both the arrays $Seed\_Budget$ and Tag_Budget require $\mathcal{O}(1)$ time. It is easy to observe that the number of times `for` loop of Line No. 8 will run for $\mathcal{O}(n)$ times. Also, the statements within this for loop will take $\mathcal{O}(1)$ time. Hence the time requirement for execution from Line No. 8 to 10 will take $\mathcal{O}(n)$ time. Also, the `for` loop of Line No. 11 will execute $\mathcal{O}(\ell)$ times. Within this loop, all the statements will take $\mathcal{O}(1)$ time. Hence, the time requirement for execution Line No. 11 to 13 is of $\mathcal{O}(\ell)$. Now, it is easy to observe that the time requirement for seed set and tag set selection from any community will depend on the number of nodes and edges that it contains, respectively. Consider the maximum number of nodes and edges of any community is denoted by n_{max} and m_{max}, respectively. So, $n_{max} = \underset{C_i \in C}{max}\ n_i$ and $m_{max} = \underset{C_i \in C}{max}\ m_i$. Now, it is easy to observe that if we analyze the time requirement for the tag selection process for the community containing m_{max} many edges and seed node selection process for the community containing n_{max} many nodes and multiply both the quantities with ℓ then we will get the time requirement for the tag set and seed set selection, respectively. First, let us consider the tag selection process. It is easy to observe that the size of T can be of $\mathcal{O}(t)$. Initializing the set T_i will take $\mathcal{O}(1)$ time. Now, as there are $\mathcal{O}(t)$ many tags, hence computing the frequency of each tag will require $\mathcal{O}(t \cdot n_{max})$ time. Now, to sort these tags based on the tag popularity value will take $\mathcal{O}(t \cdot \log t)$ time. Now, the `for` loop of Line No. 19 will take $\mathcal{O}(t)$ times. There are two conditions in the `if` statement. To check the first condition it will take $\mathcal{O}(t)$ time, however the second condition will take $\mathcal{O}(1)$ time. For the community having m_{max} many edges, the time requirement for the tag selection will take $\mathcal{O}(n_{max} \cdot t + t \cdot \log t + t^2)$ time which is reduced to $\mathcal{O}(n_{max} \cdot t + t^2)$. As mentioned previously, the time requirement for the tag selection for the whole network will take $\mathcal{O}(\ell \cdot (n_{max} \cdot t + t^2))$ time. Once the tag selection process is done, the nest step is to aggregate the tags to obtain the single influence probability and this will take $\mathcal{O}(m \cdot t)$ time. In the worst case the size of the seed set could be of $\mathcal{O}(n_{max})$. By extending the analysis of Algorithm 1, we can observe that the time requirement for seed set selection for the whole network will be $\mathcal{O}(\ell \cdot k_{max} \cdot n_{max}^2(n_{max} + m_{max}))$. So, the total time requirement

of Algorithm 2 will be $\mathcal{O}(n \log n + n + n + \ell + \ell \cdot (n_{max} \cdot t + t^2) + \ell \cdot k_{max} \cdot n_{max}^2(n_{max} + m_{max}))$. It is easy to observe that this quantity can be reduced to $\mathcal{O}(n \log n + \ell \cdot (n_{max} \cdot t + t^2) + \ell \cdot k_{max} \cdot n_{max}^2(n_{max} + m_{max}))$. Now, we can observe that ℓ can be of $\mathcal{O}(t)$ in the worst case, k_{max} and n_{max} can be of $\mathcal{O}(n)$, and m_{max} can be of $\mathcal{O}(m)$. Hence, in the worst case total time requirement will be of $\mathcal{O}(n \log n + \ell \cdot (n \cdot t + t^2) + \ell \cdot n \cdot n^2(n + m)) = \mathcal{O}(\ell \cdot (n \cdot t + t^2) + \ell \cdot n^3 \cdot (n + m))$.

Now, it is easy to observe that the extra space consumed by Algorithm 2 is to store the seed set and tag set which can be of $\mathcal{O}(n)$ and $\mathcal{O}(t)$, respectively. The arrays *Community*, *Seed_Budget*, *Tag_Budget*, and *Community_Size* will take $\mathcal{O}(n)$, $\mathcal{O}(\ell)$, $\mathcal{O}(\ell)$, and $\mathcal{O}(\ell)$ space, respectively. Also, it is easy to observe that to store the arrays T_i, *Tag_Popularity*, and *Sorted_Tags* will take $\mathcal{O}(n)$, $\mathcal{O}(t)$, and $\mathcal{O}(t)$ space, respectively. To store the aggregated influence probability for all the edges, we need $\mathcal{O}(m)$ space. Finally, we need to have $\mathcal{O}(n)$ space to store the marginal gain of the nodes. So the total space requirement by Algorithm 2 is of $\mathcal{O}(n + t + \ell + m)$. Hence, Theorem 2 holds.

Theorem 2. *The time and space requirement of Algorithm 2 will be of* $\mathcal{O}(\ell \cdot (n \cdot t + t^2) + \ell \cdot n^3 \cdot (n + m))$ *and* $\mathcal{O}(n + t + \ell + m)$*, respectively.*

4 Experimental Validation

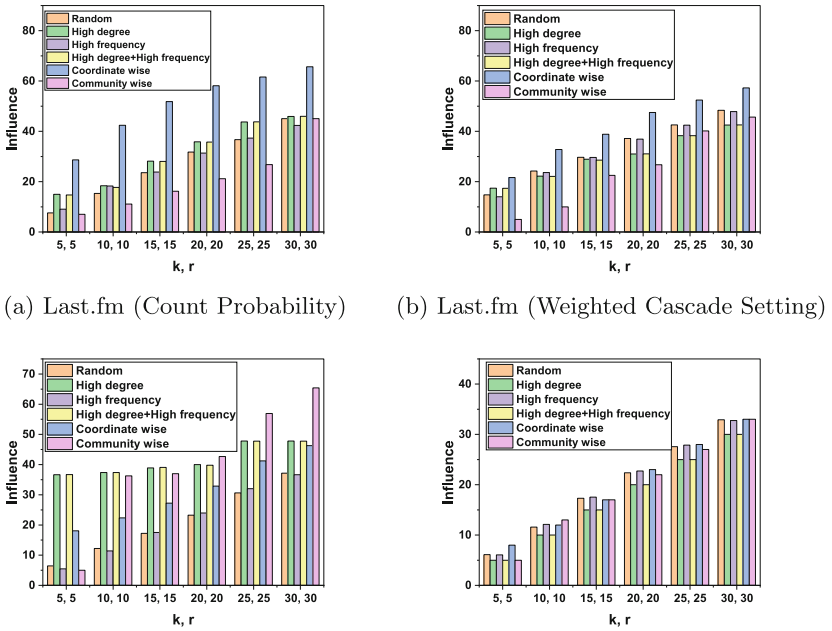
In this section, we describe the experimental evaluation of the proposed solution approaches. Initially, we start by describing the datasets.

Datasets. In this study, we have used the following two datasets Last.fm, Delicious. Last.fm contains the social relations among the listeners of this online site. Delicious is a social book-marking web service for storing, sharing, and discovering web bookmarks. These datasets have been previously used by many researchers in the domain of social networks and recommender systems.

Experimental Set Up. In this study the following parameter values needs to be set up: Influence Probability, and the value of k and r. We have considered the following two probability setting, namely, count, and weighted cascade. In count probability setting, for every tag we compute its frequency for every user of the network. Consider two users u_i and u_j and one tag t_x. Their respective frequencies are $f_{u_i}(t_x)$ and $f_{u_j}(t_x)$, respectively. The influence probability for the edge $(u_i u_j)$ for the tag t_x under the count probability setting will be $\frac{|f_{u_i}(t_x) - f_{u_j}(t_x)| + 1}{f_{u_i}(t_x)}$. In the weighted cascade setting, this probability will be $\frac{1}{f_{u_j}(t_x)}$. In this study, we consider the following (k, r) value airs: $(5, 5)$, $(10, 10)$, $(15, 15)$, $(20, 20)$, $(25, 25)$, and $(30, 30)$.

Results and Discussions. Now, we describe the experimental results. Figure 1 shows the seed node-Tag Set budget pair Vs. Influence plots for Last.fm and Delicious dataset for two different probability settings, namely Count and Weighted Cascade. From this figure, we can observe that for most of the (k, r) pair values the seed and tag set selected by our proposed solution approaches lead to more

influence compared to the baseline methods. As an example, for the Last.fm dataset with the weighted cascade setting, when the value of both k and r is set to 30, among the baseline methods, the Random method leads to the highest amount of influence and its value is 48.35. Between the proposed solution approaches, the seed and tag set selected by the community-based approach leads to the highest influence vale which is 57.29. Similar observations is made for the other probability settings as well. For the count probability setting, when the value of both k and r are set to 30, among the baseline methods, the seed and tag set selected by the high degree node-high frequency tag method leads to the influence value of 45.95. Between the two proposed methods, the seed node and tag set selected by the Coordinate-wise Method lead to the maximum value which is 65.65. Also, we observe that the computational time requirement is affordable. Hence, the proposed solution approaches lead to more amount of influence compared to baseline methods using reasonable computational time.



(a) Last.fm (Count Probability) (b) Last.fm (Weighted Cascade Setting)

(c) Delicious (Count Probability) (d) Delicious (Weighted Cascade Setting)

Fig. 1. (k, r) Pair Value Vs. Influence value for Last.fm and Delicious Dataset for Count and Weighted Cascade Probability Setting

5 Concluding Remarks

In this paper, we have studied the Tag-Based Influence Maximization Problem. First, we have shown that the tag-based influence function follows the bi-

monotonicity and bi-submodularity properties. Subsequently, we have proposed two solution methodologies with a detailed analysis. Several experiments have been conducted with real-world datasets. Our future work in this study will remain concentrated on efficient pruning techniques.

References

1. Alghamdi, R., Bellaiche, M.: A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks. Comput. Secur. **125**, 103014 (2023)
2. Ando, K., Fujishige, S., Naitoh, T.: A characterization of bisubmodular functions. Discret. Math. **148**(1–3), 299–303 (1996)
3. Banerjee, S., Pal, B.: Budgeted influence and earned benefit maximization with tags in social networks. Soc. Netw. Anal. Min. **12**(1), 21 (2022)
4. Banerjee, S., Pal, B., Jenamani, M.: Budgeted influence maximization with tags in social networks. In: Huang, Z., Beek, W., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2020. LNCS, vol. 12342, pp. 141–152. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62005-9_11
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
6. Guille, A., Hacid, H., Favre, C., Zighed, D.A.: Information diffusion in online social networks: a survey. ACM SIGMOD Rec. **42**(2), 17–28 (2013)
7. Ke, X., Khan, A., Cong, G.: Finding seeds and relevant tags jointly: for targeted influence maximization in social networks. In: Proceedings of the 2018 International Conference on Management of Data, pp. 1097–1111 (2018)
8. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
9. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Phys. Rev. Lett. **86**(14), 3200 (2001)
10. Shukla, A., Gullapuram, S.S., Katti, H., Kankanhalli, M., Winkler, S., Subramanian, R.: Recognition of advertisement emotions with application to computational advertising. IEEE Trans. Affect. Comput. **13**(2), 781–792 (2020)
11. Singh, A., Guillory, A., Bilmes, J.: On bisubmodular maximization. In: Artificial Intelligence and Statistics, pp. 1055–1063. PMLR (2012)
12. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Sci. Rep. **9**(1), 5233 (2019)
13. Zhang, Z., Shi, Y., Willson, J., Du, D.Z., Tong, G.: Viral marketing with positive influence. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–8. IEEE (2017)