Budgeted Influence Maximization with Tags in Social Networks

Suman Banerjee[1](✉), Bithika Pal[2], and Mamata Jenamani[2]

[1] Indian Institute of Technology Gandhinagar, Gandhinagar, India
suman.b@iitgn.ac.in
[2] Indian Institute of Technology Kharagpur, Kharagpur, India
bithikapal@iitkgp.ac.in, mj@iem.iitkgp.ac.in

Abstract. Given a social network, where each user is associated with a selection cost, the problem of *Budgeted Influence Maximization* (*BIM Problem*) asks to choose a subset of them (known as seed users) within the allocated budget whose initial activation leads to the maximum number of influenced nodes. In reality, the influence probability between two users depends upon the context (i.e., tags). However, existing studies on this problem do not consider the tag specific influence probability. To address this issue, in this paper we introduce the Tag-Based Budgeted Influence Maximization Problem (*TBIM Problem*), where along with the other inputs, a tag set (each of them is also associated with a selection cost) is given, each edge of the network has the tag specific influence probability, and here the goal is to select influential users as well as influential tags within the allocated budget to maximize the influence. Considering the fact that different tag has different popularity across the communities of the same network, we propose three methodologies that work based on *effective marginal influence gain computation*. The proposed methodologies have been analyzed for their time and space requirements. We evaluate the methodologies with three datasets, and observe, that these can select seed nodes and influential tags, which leads to more number of influenced nodes compared to the baseline methods.

Keywords: Social network · BIM problem · Seed nodes · Tags

1 Introduction

A social network is an interconnected structure among a group of agents. One of the important phenomenon of social networks is the diffusion of information [5]. Based on the diffusion process, a well studied problem is the *Social Influence Maximization* (*SIM Problem*), which has an immediate application in the context of *viral marketing*. The goal here is to get wider publicity for a product by initially distributing a limited number of free samples to highly influential users. For a

The work of the first author is supported by the Institute Post Doctoral Fellowship Grant of IIT Gandhinagar (MIS/IITGN/PD-SCH/201415/006).

Z. Huang et al. (Eds.): WISE 2020, LNCS 12342, pp. 141–152, 2020.
https://doi.org/10.1007/978-3-030-62005-9_11

given social network and a positive integer k, the SIM Problem asks to select k users for initial activation to maximize the influence in the network. Due to potential application of this problem in viral marketing [4], different solution methodologies have been developed. Look into [2] for recent survey.

Recently, a variant of this problem has been introduced by Nguyen and Zheng [7], where the users of the network are associated with a selection cost and the seed set selection is to be done within an allocated budget. There are a few approaches to solve the problem [1,7]. In all these studies, it is implicitly assumed that irrespective of the context, influence the probability between two users will be the same, i.e., there is a single influence probability associated with every edge. However, in reality, the scenario is different. It is natural that a sportsman can influence his friend in any sports related news with more probability compared to political news. This means the influence probability between any two users are context specific, and hence, in Twitter a follower will re-tweet if the tweet contains some specific hash tags. To address this issue, we introduce the *Tag-based Budgeted Influence Maximization (TBIM) Problem*, which considers the tag specific influence probability assigned with every edge.

Ke et al. [6] studied the problem of finding k seed nodes and r influential tags in a social network. However, their study has two drawbacks. First, in reality most of the social networks are formed by rational human beings. Hence, once a node is selected as seed then incentivization is required (e.g., free or discounted sample of the item to be advertised). Also, the cost of displaying a viral marketing message in any media platforms such as *Vonag*[1] is associated with a cost. As the message is constituted by the tags, hence it is important to consider the individual cost for tags. Their study does not consider these issues. Secondly, in their study, they have done the tag selection process at the network level. However, in reality, popular tags may vary from one community to another in the same network. Figure 1 shows community wise distribution of Top 5 tags for the Last.fm dataset. From the figure, it is observed that Tag No. 16 has the highest popularity in Community 3. However, its popularity is very less in Community 4. This signifies that tag selection in network level may not be always helpful to spread influence in each community of the network. To mitigate this issues, we propose three solution methodologies for the TBIM Problem, where the tag selection is done community wise. To the best of our knowledge, this is the second study in this direction. The main contributions of this paper are as follows:

- Considering the tag specific influence probability, in this paper we introduce the Tag-based Budgeted Influence Maximization Problem (TBIM Problem).
- Two iterative methods have been proposed with their detailed analysis.
- To increase the scalability, an efficient pruning technique has been developed.
- The proposed methodologies have been implemented with three publicly available datasets and a set of experiments have been performed to show their effectiveness.

[1] https://www.vonage.com.

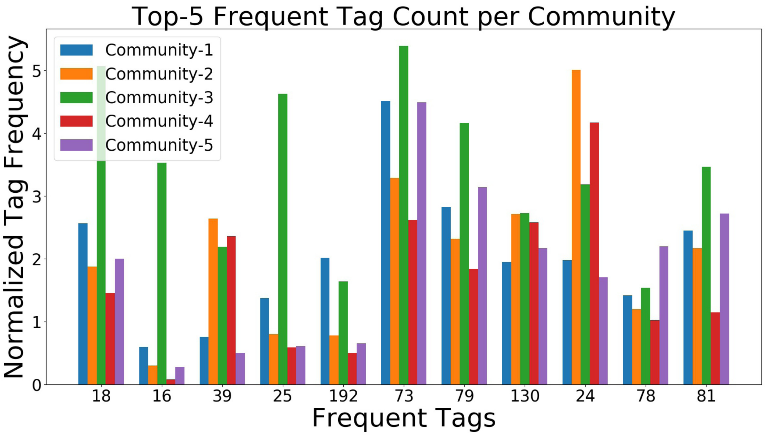


Fig. 1. Community specific distribution top 5 tags in 'Last.fm' dataset

Rest of the paper is arranged as follows: Sect. 2 contains some background material and defines the TBIM Problem formally. The proposed solution methodologies for this problem have been described in Sect. 3. In Sect. 4, we report the experimental results of the proposed methodologies.

2 Background and Problem Definition

The social network is represented as a directed and edge weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P})$, where the vertex set, $\mathcal{V}(\mathcal{G}) = \{u_1, u_2, \dots, u_n\}$ is the set of n users, the edge set $\mathcal{E}(\mathcal{G}) = \{e_1, e_2, \dots, e_m\}$ is the set of m social ties among the users. Along with $\mathcal{G}$, we are also given with a tag set $T = \{t_1, t_2, \dots, t_a\}$ relevant to the users of the network. $\mathcal{P}$ is the edge weight function that assigns each edge to its tag specific influence probability, i.e., $\mathcal{P} : \mathcal{E}(\mathcal{G}) \longrightarrow (0, 1]^{|T|}$. This means that each edge of the network is associated with a influence probability vector, whose each dimension is for a particular tag. For all $(u_i u_j) \in \mathcal{E}(\mathcal{G})$, we denote its corresponding influence probability vector as $\mathcal{P}_{u_i \rightarrow u_j}$. Also, for a particular tag $t \in T$ and an edge $(u_i u_j) \in \mathcal{E}(\mathcal{G})$, we denote the influence probability of the edge $(u_i u_j)$ for the tag t as $\mathcal{P}^t_{u_i \rightarrow u_j}$. Now, a subset of the available tags $T^{'} \subseteq T$ which are relevant to the campaign may be used. It is important how to compute the effective probability for each edge and this depends upon how the selected tags are aggregated. In this study, we perform the *independent tag aggregation* shown in Eq. 1.

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

Diffusion in Social Networks. To conduct a campaign using a social network, a subset of the users need to be selected initially as seed nodes (denoted by $\mathcal{S}$). The users in the set $\mathcal{S}$ are informed initially, and the others are ignorant about the information. These seed users start the diffusion, and the information is diffused by the rule of an information diffusion model. There are many such rules proposed in the literature. One of them is the MIA Model [8]. Recently, this model has been used by many existing studies in influence maximization [9]. Now, we state a few preliminary definitions.

Definition 1 (Propagation Probability of a Path). *Given two vertices $u_i, u_j \in V(\mathcal{G})$, let $\mathbb{P}(u_i, u_j)$ denotes the set of paths from the vertex u_i to u_j. For any arbitrary path $p \in \mathbb{P}(u_i, u_j)$ the propagation probability is defined as the product of the influence probabilities of the constituent edges of the path.*

$$\mathcal{P}(p) = \begin{cases} \prod\limits_{(u_i u_j) \in \mathcal{E}(p)} \mathcal{P}^{T^{'}}_{u_i \rightarrow u_j} & \textit{if } \mathbb{P}(u_i, u_j) \neq \phi \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

Here, $\mathcal{E}(p)$ denotes the edges that constitute the path p.

Definition 2 (Maximum Probabilistic Path). *Given two vertices $u_i, u_j \in \mathcal{V}(\mathcal{G})$, the maximum probabilistic path is the path with the maximum propagation probability and denoted as $p^{max}_{(u_i u_j)}$. Hence,*

$$p^{max}_{(u_i u_j)} = \underset{p \in \mathbb{P}(u_i, u_j)}{argmax} \ \mathcal{P}(p) \tag{3}$$

Definition 3 (Maximum Influence in Arborescence). *For a given threshold θ, the maximum influence in-arborescence of a node v is defined as*

$$MIIA(v, \theta) = \bigcup_{u \in \mathcal{V}(\mathcal{G}), \mathcal{P}(p^{max}_{(uv)}) \geq \theta} p^{max}_{(uv)} \tag{4}$$

Given a seed set $\mathcal{S}$ and a node $v \notin \mathcal{S}$, in MIA Model, the influence from $\mathcal{S}$ to v is approximated by the rule that for any $u \in \mathcal{S}$ can influence v through the paths in $p^{max}_{(uv)}$. The influence probability of a node $u \in MIIA(v, \theta)$ is denoted as $ap(u, \mathcal{S}, MIIA(v, \theta))$, which is the probability that the node u will be influenced by the nodes in $\mathcal{S}$ and influence is propagated through the paths in $MIIA(v, \theta)$. This can be computed by the Algorithm 2 of [8]. Hence, the influence spread obtained by the seed set $\mathcal{S}$ is given by the Eq. 5.

$$\sigma(\mathcal{S}) = \sum_{v \in \mathcal{V}(\mathcal{G})} ap(v, \mathcal{S}, MIIA(v, \theta)) \tag{5}$$

Problem Definition. To study the BIM Problem along with the input social network, we are given with the selection costs of the users which is characterized by the cost function $\mathcal{C} : \mathcal{V}(\mathcal{G}) \rightarrow \mathbb{R}^{+}$, and a fixed budget $\mathcal{B}$. For any user $u \in \mathcal{V}(\mathcal{G})$, its selection cost is denoted as $\mathcal{C}(u)$. The BIM Problem asks to choose a subset of users $\mathcal{S}$ such that $\sigma(\mathcal{S})$ is maximized and $\sum\limits_{u \in \mathcal{S}} \mathcal{C}(u) \leq \mathcal{B}$. Let, $\mathcal{K} = \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_\ell\}$ denotes the set of communities of the network. Naturally, all the tags that are considered in a specific context (i.e., T) may not be relevant to each of the communities. We denote the relevant tags of the community $\mathcal{K}_i$ as $T_{\mathcal{K}_i}$. It is important to observe that displaying a tag in any on-line platform may associate some cost, which can be characterized by the tag cost function $\mathcal{C}^T : T \rightarrow \mathbb{R}^+$. Now, for a set of given tags and seed nodes what will be the number of influenced nodes in the network? This can be defined as the tag-based influence function. For a given seed set $\mathcal{S}$ and tag set $T^{'}$, the tag-based influence function $\sigma(\mathcal{S}, T^{'})$ returns the number of influenced nodes, which is defined next.

Definition 4 (Tag-Based Influence Function). *Given a social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P})$, a seed set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, tag set $T^{'} \subseteq T$, the tag-based influence function $\sigma^{\mathcal{T}}$ that maps each combination of subset of the nodes and tags to the number of influenced nodes, i.e., $\sigma^{\mathcal{T}} : 2^{\mathcal{V}(\mathcal{G})} \times 2^{T} \longrightarrow \mathbb{R}_0$.*

Finally, we define the Tag-based Budgeted Influence Maximization Problem.

Definition 5 (TBIM Problem). *Given a social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P})$, Tag set T, seed cost function $\mathcal{C}^S : \mathcal{V}(\mathcal{G}) \rightarrow \mathbb{R}^+$, tag cost function $\mathcal{C}^T : T \rightarrow \mathbb{R}^+$ and the budget $\mathcal{B}$, the TBIM Problem asks to select a subset of the tags from the communities, i.e., $T^{'}_{\mathcal{K}_i} \subseteq T_{\mathcal{K}_i}$, $\forall i \in [\ell]$ (here, $T^{'}_{\mathcal{K}_i} \cap T^{'}_{\mathcal{K}_j} = \emptyset$, $\forall i \neq j$ and $T^{'} = \bigcup\limits_{i \in [\ell]} T_{\mathcal{K}_i}$), and nodes $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$ to maximize $\sigma^{\mathcal{T}}(\mathcal{S}, T^{'})$ such that $\sum\limits_{u \in \mathcal{S}} \mathcal{C}^S(u) + \sum\limits_{i \in [\ell]} \sum\limits_{t \in T_{\mathcal{K}_i}} \mathcal{C}^T(t) \leq \mathcal{B}$.*

3 Proposed Methodologies

Here, we describe two different approaches and one subsequent improvement to select tags and seed users for initiating the diffusion process. Before stating the proposed approaches, we first define the *Effective Marginal Influence Gain.*

Definition 6 (Effective Marginal Influence Gain). *Given a seed set $\mathcal{S} \subset \mathcal{V}(\mathcal{G})$, tag set $T^{'} \subset T$, the effective marginal influence gain (EMIG) of the node $v \in \mathcal{V}(\mathcal{G}) \setminus \mathcal{S}$ (denoted as δ_v) with respect to the seed set $\mathcal{S}$ and tag set $T^{'}$ is defined as the ratio between the marginal influence gain to its selection cost, i.e.,*

$$\delta_v = \frac{\sigma^{\mathcal{T}}(\mathcal{S} \cup \{v\}, T^{'}) - \sigma^{\mathcal{T}}(\mathcal{S}, T^{'})}{\mathcal{C}^S(v)}. \tag{6}$$

In the similar way, for any tag $t \in T \setminus T^{'}$, its EMIG is defied as

$$\delta_t = \frac{\sigma^{\mathcal{T}}(\mathcal{S}, T^{'} \cup \{t\}) - \sigma^{\mathcal{T}}(\mathcal{S}, T^{'})}{\mathcal{C}^T(t)}. \tag{7}$$

For the user-tag pair (v, t), $v \in \mathcal{V}(\mathcal{G}) \setminus \mathcal{S}$, and $t \in T \setminus T^{'}$, its EMIG is defined as

$$\delta_{(v,t)} = \frac{\sigma^{\mathcal{T}}(\mathcal{S} \cup \{v\}, T^{'} \cup \{t\}) - \sigma^{\mathcal{T}}(\mathcal{S}, T^{'})}{\mathcal{C}^S(v) + \mathcal{C}^T(t)}. \tag{8}$$

3.1 Methodologies Based on Effective Marginal Influence Gain Computation of User-Tag Pairs (EMIG-UT)

In this method, first the community structure of the network is detected, and the total budget is divided among the communities based on its size. In each community, the shared budget is divided into two halves to be utilized to select tags and seed nodes, respectively. Next, we sort the communities based on its size in ascending order. Now, we take the smallest community first and select the most frequent tag which is less than or equal to the budget. Next, each community from smallest to the largest is processed for tag and seed node selection in the following way. Until the budget for both tag and seed node selection is exhausted, in each iteration the user-tag pair that causes maximum EMIG value is chosen and kept into the seed set and tag set, respectively. The extra budget is transferred to the largest community. Algorithm 1 describe the procedure.

Algorithm 1: Effective Marginal Influence Gain Computation of User-Tag Pairs (EMIG-UT)

```
Data: Social Network $\mathcal{G}$, Tag Set $T$, Node Cost Function $\mathcal{C}^S$, User-Tag Count
      Matrix $\mathcal{M}$, Tag Cost Function $\mathcal{C}^T$, Budget $\mathcal{B}$
Result: Seed set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, and Tag set $T^{'} \subseteq T$
$\mathcal{S} = \emptyset$; $T^{'} = \emptyset$; $Community = Community\_Detection(\mathcal{G})$;
$\mathcal{K} \longleftarrow \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_l\}$; $\mathcal{K}_{max} = Largest\_Community(\mathcal{K})$;
$T_{\mathcal{K}} = Crete\_Matrix(|\mathcal{K}|, |T|, 0)$;
$T_{\mathcal{K}} =$ Count the tag frequencies in each communities;
Sort row of $T_{\mathcal{K}}$corresponding to the smallest community ;
$Create\_Vector(\mathcal{B}^k_{\mathcal{S}}, \ell, 0)$; $Create\_Vector(\mathcal{B}^k_T, \ell, 0)$;
for $i = 1$ $to$ $|\mathcal{K}|$ do
    $\mathcal{B}^k = \frac{|\mathcal{V}_{\mathcal{K}_i}|}{n}.\mathcal{B}$; $\mathcal{B}^k_{\mathcal{S}}[i] = \frac{\mathcal{B}^k}{2}$; $\mathcal{B}^k_T[i] = \frac{\mathcal{B}^k}{2}$;
end
$\mathcal{K} = Sort(\mathcal{K})$; $T^{'} = T^{'} \cup \{t^{'} : t^{'}$ is most frequent in $\mathcal{K}_1$ and $\mathcal{C}(t) \leq \mathcal{B}^k_T[i]\}$;
$\mathcal{B}^k_T[1] = \mathcal{B}^k_T[1] - \mathcal{C}^T(t^{'})$;
for $i = 1$ $to$ $|\mathcal{K}|$ do
    while $\mathcal{B}^k_{\mathcal{S}}[i] > 0$ $and$ $\mathcal{B}^k_T[i] > 0$ do
        $(u,t) = \underset{\substack{v \in \mathcal{V}_{\mathcal{K}_i}(\mathcal{G}) \setminus \mathcal{S}, \mathcal{C}(v) \leq \mathcal{B}^k_{\mathcal{S}}[i]; \\ t^{''} \in T \setminus T^{'}, \mathcal{C}(t^{''}) \leq \mathcal{B}^k_T[i]}}{argmax} \delta_{(v,t^{''})}$; $\mathcal{S} = \mathcal{S} \cup \{u\}; T^{'} = T^{'} \cup \{t\}$;
        $\mathcal{B}^k_{\mathcal{S}}[i] = \mathcal{B}^k_{\mathcal{S}}[i] - \mathcal{C}^S(u); \mathcal{B}^k_T[i] = \mathcal{B}^k_T[i] - \mathcal{C}^T(t)$;
    end
    $\mathcal{B}^k_{\mathcal{S}}[max] = \mathcal{B}^k_{\mathcal{S}}[max] + \mathcal{B}^k_{\mathcal{S}}[i]; \mathcal{B}^k_T[max] = \mathcal{B}^k_T[max] + \mathcal{B}^k_{\mathcal{S}}[i]$;
end
```

Now, we analyze Algorithm 1. For detecting communities using Louvian Method requires $\mathcal{O}(n \log n)$ time. Computing the tag count in each communities requires $\mathcal{O}(n|T|)$ time. `Community` is the array that contains the community number of the user in which they belong to, i.e., `Community[i]=x` means the user u_i belongs to Community $\mathcal{K}_x$. From this array, computing the size of each communities and finding out the maximum one requires $\mathcal{O}(n)$ time. Dividing the budget among the communities for seed node and tag selection and Sorting the communities require $\mathcal{O}(\ell)$ and $\mathcal{O}(\ell \log \ell)$ time, respectively. From the smallest community, choosing the highest frequency tag requires $\mathcal{O}(|T| \log |T|)$ time. Time requirement for selecting tags and seed nodes in different communities will be different. For any arbitrary community $\mathcal{K}_x$, let, $\mathcal{C}(\mathcal{S}^{min}_{\mathcal{K}_x})$ and $\mathcal{C}(T^{min}_{\mathcal{K}_x})$ denote the minimum seed and tag selection cost of this community, respectively. Hence, $\mathcal{C}(\mathcal{S}^{min}_{\mathcal{K}_x}) = \underset{u \in V_{\mathcal{K}_x}}{min}\, \mathcal{C}^{S}(u)$ and $\mathcal{C}(T^{min}_{\mathcal{K}_x}) = \underset{t \in T_{\mathcal{K}_x}}{min}\, \mathcal{C}^{T}(t)$. Here, $V_{\mathcal{K}_x}$ and $T_{\mathcal{K}_x}$ denote the nodes and relevant tags in community $\mathcal{K}_x$, respectively. Also, $\mathcal{B}^{\mathcal{S}}_{\mathcal{K}_x}$ and $\mathcal{B}^{T}_{\mathcal{K}_x}$ denotes the budget for selecting seed nodes and tags for the community $\mathcal{K}_x$, respectively. Now, it can be observed that, the number of times `while` loop (Line number 13 to 16) runs for the community $\mathcal{K}_x$ is $min\{\frac{\mathcal{B}^{\mathcal{S}}_{\mathcal{K}_x}}{\mathcal{C}(\mathcal{S}^{min}_{\mathcal{K}_x})}, \frac{\mathcal{B}^{T}_{\mathcal{K}_x}}{\mathcal{C}(T^{min}_{\mathcal{K}_x})}\}$ and it is denoted as $r_{\mathcal{K}_x}$. Let, $r_{max} = \underset{\mathcal{K}_x \in \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_\ell\}}{max} r_{\mathcal{K}_x}$. The number of times the marginal influence gain needs to be computed is of $\mathcal{O}(\ell r_{max} n |T|)$. Assuming the time requirement for computing the MIIA for a single node with threshold θ is of $\mathcal{O}(t_\theta)$ [8]. Hence, computation of $\sigma(\mathcal{S})$ requires $\mathcal{O}(n t_\theta)$ time. Also, after updating the tag set in each iteration updating the aggregated influence probability requires $\mathcal{O}(m)$ time. Hence, execution from Line 17 to 24 of Algorithm 1 requires $\mathcal{O}(\ell r_{max} n |T| (n t_\theta + m))$. Hence, the total time requirement for Algorithm 1 of $\mathcal{O}(n \log n + n|T| + n + \ell + \ell \log \ell + \ell r_{max} n |T| (n t_\theta + m)) = \mathcal{O}(n \log n + \ell \log \ell + \ell r_{max} n |T| (n t_\theta + m))$. Additional space requirement for Algorithm 1 is to store the `Community` array which requires $\mathcal{O}(n)$ space, for $\mathcal{B}^k_{\mathcal{S}}$ and $\mathcal{B}^k_{T}$ require $\mathcal{O}(\ell)$, for $T_{\mathcal{K}}$ requires $\mathcal{O}(\ell |T|)$, for storing MIIA path $\mathcal{O}(n(n_{i\theta} + n_{o\theta}))$ [8], for aggregated influence probability $\mathcal{O}(m)$, for $\mathcal{S}$ and $T^{'}$ require $\mathcal{O}(n)$ and $\mathcal{O}(|T^{'}|)$, respectively. Formal statement is presented in Theorem 1.

Theorem 1. *Time and space requirement of Algorithm 1 is of* $\mathcal{O}(n \log n + |T| \log |T| + \ell \log \ell + \ell r_{max} n |T| (n t_\theta + m))$ *and* $\mathcal{O}(n(n_{i\theta} + n_{o\theta}) + \ell |T| + m)$, *respectively.*

3.2 Methodology Based on Effective Marginal Influence Gain Computation of Users (EMIG-U)

As observed in the experiments, computational time requirement of Algorithm 1 is very high. To resolve this problem, Algorithm 2 describes the *Effective Marginal Influence Gain Computation of Users (EMIG-U)* approach, where after community detection and budget distribution, high frequency tags from the communities are chosen (Line 3 to 11), and effective influence probability for each of the edges are computed. Next, from each of the communities until their respective budget is exhausted, in each iteration the node that causes maximum EMIG value are chosen as seed nodes. As described previously, time requirement

Algorithm 2: Effective Marginal Influence Gain Computation of User (EMIG-U)

```
   Data: Social Network $\mathcal{G}$, Tag Set $T$, Node Cost Function $\mathcal{C}^S$, User-Tag Count
         Matrix $\mathcal{M}$, Tag Cost Function $\mathcal{C}^T$, Budget $\mathcal{B}$
   Result: Seed set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, and Tag set $T^{'} \subseteq T$
 1 Execute Line Number 1 to 14 of Algorithm 1;
 2 Sort each row of the $T_{\mathcal{K}}$ matrix;
 3 for $i = 1$ to $|\mathcal{K}|$ do
 4     for $j = 1$ to $|T|$ do
 5         if $\mathcal{C}(t_j) \leq \mathcal{B}_T^k[i]$ and $t_j \notin T^{'}$ then
 6             $T^{'} = T^{'} \cup \{t_j\}$; $\mathcal{B}_T^k[i] = \mathcal{B}_T^k[i] - \mathcal{C}^T(t_j)$;
 7         end
 8     end
 9     $\mathcal{B}_T^k[max] = \mathcal{B}_T^k[max] + \mathcal{B}_T^k[i]$;
10 end
11 for All $(u_i u_j) \in \mathcal{E}(\mathcal{G})$ do
12     Compute aggregated influence using Equation 1;
13 end
14 for $i = 1$ to $|\mathcal{K}|$ do
15     while $\mathcal{B}_{\mathcal{S}}^k[i] > 0$ do
16         $u = \underset{v \in \mathcal{V}_{\mathcal{K}_i}(\mathcal{G}) \setminus \mathcal{S}, \mathcal{C}(v) \leq \mathcal{B}_{\mathcal{S}}^k[i]}{argmax} \delta_v$; $\mathcal{S} = \mathcal{S} \cup \{u\}$;
17         $\mathcal{B}_{\mathcal{S}}^k[i] = \mathcal{B}_{\mathcal{S}}^k[i] - \mathcal{C}^S(u)$;
18     end
19     $\mathcal{B}_{\mathcal{S}}^k[max] = \mathcal{B}_{\mathcal{S}}^k[max] + \mathcal{B}_{\mathcal{S}}^k[i]$;
20 end
```

for executing Line 1 to 14 is $\mathcal{O}(n \log n + |T| \log |T| + n|T| + n + \ell + \ell \log \ell) = \mathcal{O}(n \log n + |T| \log |T| + n|T| + \ell \log \ell)$. Sorting each row of the matrix $T_{\mathcal{K}}$ requires $\mathcal{O}(\ell |T| \log |T|)$ time. For any arbitrary community $\mathcal{K}_x$, the number of times the `for` loop will run in the worst case is of $\mathcal{O}(\frac{\mathcal{B}_{\mathcal{K}_x}^T}{\mathcal{C}(T_{\mathcal{K}_x}^{min})})$. Let, $t_{max} = \underset{\mathcal{K}_x \in \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_\ell\}}{max} \frac{\mathcal{B}_{\mathcal{K}_x}^T}{\mathcal{C}(T_{\mathcal{K}_x}^{min})}$. Also, in every iteration, it is to be checked whether the selected tag is already in $T^{'}$ or not. Hence, the worst case running time from Line 3 to 11 will be of $\mathcal{O}(\ell t_{max} |T^{'}|)$. Computing aggregated influence probabilities for all the edges (Line 12 to 14) requires $\mathcal{O}(m|T^{'}|)$ time. For the community $\mathcal{K}_x$, the number of times the `while` loop in Line 16 will run in worst case is of $\mathcal{O}(\frac{\mathcal{B}_{\mathcal{K}_x}^{\mathcal{S}}}{\mathcal{C}(\mathcal{S}_{\mathcal{K}_x}^{min})})$. Let, $s_{max} = \underset{\mathcal{K}_x \in \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_\ell\}}{max} \frac{\mathcal{B}_{\mathcal{K}_x}^{\mathcal{S}}}{\mathcal{C}(\mathcal{S}_{\mathcal{K}_x}^{min})}$. Hence, the number of times the marginal influence gain will be computed is of $\mathcal{O}(\ell s_{max} n)$. Contrary to Algorithm 1, in this case MIIA path needs to be computed only once after the tag probability aggregation is done. Hence, worst case running time from Line 15 to 22 is of $\mathcal{O}(\ell s_{max} n + n t_\theta)$. The worst case running time of Algorithm 2 is of $\mathcal{O}(n \log n +$

$|T|\log|T| + n|T| + \ell\log\ell + \ell|T|\log|T| + \ell t_{max}|T^{'}| + m|T^{'}| + \ell s_{max}n + nt_{\theta}) = \mathcal{O}(n\log n + n|T| + \ell\log\ell + \ell|T|\log|T| + \ell t_{max}|T^{'}| + m|T^{'}| + \ell s_{max}n + nt_{\theta})$. It is easy to verify that the space requirement of Algorithm 2 will be same as Algorithm 1. Hence, Theorem 2 holds.

Theorem 2. *Running time and space requirement of Algorithm 2 is of* $\mathcal{O}(n\log n + n|T| + \ell\log\ell + \ell|T|\log|T| + \ell t_{max}|T^{'}| + m|T^{'}| + \ell s_{max}n + nt_{\theta})$ *and* $\mathcal{O}(n(n_{i\theta} + n_{o\theta}) + \ell|T| + m)$, *respectively.*

3.3 Efficient Pruning Technique (EMIG-U-Pru)

Though, Algorithm 2 has better scalability compared to Algorithm 1, still it is quite huge. The main performance bottleneck of Algorithm 2 is the excessive number of EMIG computations. Hence, it will be beneficial, if we can prune off some of the nodes, in such a way that even if we don't perform this computation for these nodes, still it does not affect much on the influence spread. We propose the following pruning strategy. Let, $\mathcal{S}^i$ denotes the seed set after the i^{th} iteration. $\forall u \in V(\mathcal{G})\setminus\mathcal{S}^i$, if the outdegree of u, i.e. $outdeg(u)$ will be decremented by $|\mathcal{N}^{in}(u) \cap \mathcal{S}^i|$, where $\mathcal{N}^{in}$ denotes the set of incoming neighbors of u. All the nodes in $V(\mathcal{G})\setminus\mathcal{S}^i$ are sorted based on the computed outdegree to cost ratio and top-k of them are chosen for the EMIG computation. We have stated for the i^{th} iteration. However, the same is performed in every iteration. Due to the space limitation, we are unable to present the entire algorithm and its analysis. However, we state the final result in Theorem 3.

Theorem 3. *Running time and space requirement of the proposed pruning strategy is of* $\mathcal{O}(n\log n + \ell|T|\log|T| + n|T| + \ell\log\ell + \ell t_{max}|T^{'}| + m|T^{'}| + \ell s_{max}(n_{max}|\mathcal{S}| + n_{max}\log n_{max} + k) + nt_{\theta})$ *and* $\mathcal{O}(n(n_{i\theta} + n_{o\theta}) + \ell|T| + m)$, *respectively.*

4 Experimental Evaluation

We use three datasets, namely, **Last.fm** [3], **Delicious** [3], and **LibraryThing** [10]. No. of nodes, edges and tags in these datasets are 1288, 11678, and 11250; 1839, 25324, and 9749; and 15557, 108987, and 17228, respectively. For all the three datasets it have been observed that the frequency of the tags decreases exponentially. Hence, instead of dealing with all the tags, we have selected 1000 tags in each datasets using most frequent tags per community. Now, we describe the experimental setup. Initially, we start with the influence probability setting.

- **Trivalency Setting**: In this case, for each edge and for all the tags influence probabilities are randomly assigned from the set $\{0.1, 0.01, 0.001\}$.
- **Count Probability Setting**: By this rule, for each edge (u_iu_j) its influence probability vector is computed as follows. First, element wise subtraction from $\mathcal{M}^{u_i}$ to $\mathcal{M}^{u_j}$ is performed. If there are some negative entries, they are changed to 0. We call the obtained vector as $\mathcal{M}^{u_i - u_j}$. Next, 1 is added

with each entries of the vector $\mathcal{M}^{u_i}$. We call this vector as $\mathcal{M}^{u_i+1}$. Now, the element wise division of $\mathcal{M}^{u_i - u_j}$ is performed by $\mathcal{M}^{u_i+1}$. The resultant vector is basically the influence probability vector for the edge $(u_j u_i)$. Here 1 is added with each of the entries of $\mathcal{M}^{u_i}$ before the division just to avoid infinite values in the influence probability vector.
- **Weighted Cascade Setting**: Let, $\mathcal{N}^{in}(u_i)$ denotes the set of incoming neighbors for the node u_i. In standard weighted cascade setting, $\forall u_j \in \mathcal{N}^{in}(u_i)$, the influence probability for the edges $(u_j u_i)$ is equal to $\frac{1}{deg^{in}(u_i)}$. Here, we have adopted this setting in a little different way. Let, $\mathcal{M}^{u_i}$ denotes the tag count vector of the user u_i (i^{th} row of the matrix $\mathcal{M}$). Now, $\forall u_j \in \mathcal{N}^{in}(u_i)$, we select the corresponding rows from $\mathcal{M}$, apply column-wise sum on the tag-frequency entries and perform the element wise division of the vector $\mathcal{M}^{u_i}$ by the summed up vector. The resultant vector is assigned as the influence probability for all the edges from $\forall u_j \in \mathcal{N}^{in}(u_i)$ to u_i.

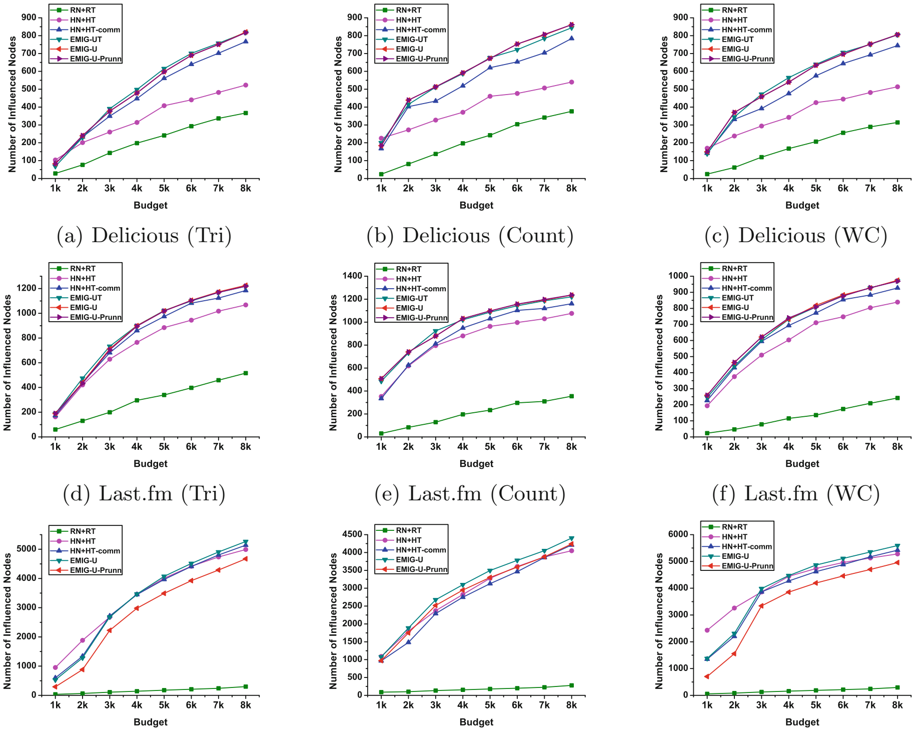
Cost and Budget. We have adopted the *random setting* for assigning selection cost to each user and tag as mentioned in [7]. Selection cost for each user and tag are selected from the intervals $[50, 100]$ and $[25, 50]$, respectively uniformly at random. We have experimented with fixed budget values starting with 1000, continued until 8000, incremented each time by 1000, i.e., $\mathcal{B} = \{1000, 2000, \ldots, 8000\}$.

The following baseline methods have been used for comparison.

- **Random Nodes and Random Tags (RN+RT):** According to this method, the allocated budget is divided into two equal halves. One half will be spent for selecting seed nodes and the other one for selecting tags. Now seed nodes and tags are chosen randomly until their respective budgets are exhausted.
- **High Degree Nodes and High Frequency Tags (HN+HT):** In this method, after dividing the budget into two equal halves, high degree nodes and high frequency tags are chosen until their respective budget is exhausted.
- **High Degree Nodes and High Frequency Tags with Communities (HN+HT+COM):** In this method, after dividing the budget into two equal halves, first the community structure of the network is detected. Both of these divided budgets are further divided among the communities based on the community size. Then apply **HN+HT** for each community.

The implementations have been done with *Python 3.5 + NetworkX 2.1* on a HPC Cluster with 5 nodes each of them having 64 cores and 160 GB of memory and available at https://github.com/BITHIKA1992/BIM_with_Tag.

Figure 2 shows the Budget vs. Influence plot for all the datasets. From the figures, it has been observed that the seed set selected by proposed methodologies leads to more influence compared to the baseline methods. For the 'Delicious' dataset, for $\mathcal{B} = 8000$, under weighted cascade setting, among the baseline methods, **HN+HT+COMM** leads to the expected influence of 744, whereas the same for **EMIG-UT**, **EMIG-U**, **EMIG-U-Prunn** methods are 804, 805, and 805, respectively, which is approximately 8% more compared to



(a) Delicious (Tri)

(b) Delicious (Count)

(c) Delicious (WC)

(d) Last.fm (Tri)

(e) Last.fm (Count)

(f) Last.fm (WC)

(g) LibraryThing (Tri)

(h) LibraryThing (Count)

(i) LibraryThing (WC)

Fig. 2. Budget Vs. influence plot for **Delicious**, **Last.fm**, and **LibraryThing** datasets under the trivalency, weighted cascade and count probability settings.

HN+HT+COMM. The influence due to the seed set selected by **EMIG-U-Prunn** under Weighted Cascade, trivalency, and, Count setting are 805, 816, and 861 which are 62.5%, 63.2% and 66.85% of the number of nodes of the network, respectively. Also, we observe that for a given budget, the number of seed nodes selected by the proposed methodologies are always more compared to baseline methods. For example, under the trivalency setting with $\mathcal{B} = 8000$, the number of seed nodes selected by **RN+RT**, **HN+HT**, and **HN+HT+COMM** methods are 54. The same for **EMIG-UT**, **EMIG-U**, and **EMIG-U-Prunn** are 59, 62, and 62, respectively.

In case of 'Last.fm' dataset also, similar observations are made. For example, under trivalency setting with $\mathcal{B} = 8000$, the expected influence by the proposed methodologies **EMIG-UT**, **EMIG-U**, and **EMIG-U-Prunn** are 1230, 1226, and 1219, respectively. The same by **RN+RT**, **HN+HT**, and **HN+HT+COMM** methods are 515, 1067, and 1184, respectively. In this dataset also, it has been observed that the number of seed nodes selected by the proposed methodologies are more compared to the baseline methods. For example, in trivalency setting, with $\mathcal{B} = 8000$, the number of seed nodes selected by **HN+HT+COMM**, and **EMIG-U-Prunn** are 51 and 59, respectively.

In LibraryThing dataset, the observations are not fully consistent with previous two datasets. It can be observed from Fig. 2 ((g), (h), and (i)) that due to the pruning, the expected influence dropped significantly. As an example, in trivalency setting, for $\mathcal{B} = 8000$, the expected influence by **EMIG-U** and **EMIG-U-Prunn** methods are 5265 and 4673, respectively. It is due to the following reason. Recall, that in the **EMIG-U-Prunn** methodology, we have only considered 200 nodes for computing marginal gain in each iteration. As this dataset is larger than previous two, hence their are many prospective nodes for which the marginal has not been computed. However, it is interesting to observe still the number of seed nodes selected by the proposed methodologies are more compared to baseline methods.

Due to space limitation, we are unable to discuss about computational time requirement. However, we mention one observation is that the ratio between the computational time of **EMIG-U** and **EMIG-U-Prunn** for the Delicious, Last.fm, and LibraryThing are approximately 1.1, 2, and 10, respectively.

References

1. Banerjee, S., Jenamani, M., Pratihar, D.K.: Combim: a community-based solution approach for the budgeted influence maximization problem. Expert Syst. Appl. **125**, 1–13 (2019)
2. Banerjee, S., Jenamani, M., Pratihar, D.K.: A survey on influence maximization in a social network. Knowl. Inf. Syst. 1–39 (2020)
3. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In: Proceedings of the 5th ACM Conference on Recommender Systems, RecSys 2011. ACM, New York (2011)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038. ACM (2010)
5. 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)
6. 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. ACM (2018)
7. Nguyen, H., Zheng, R.: On budgeted influence maximization in social networks. IEEE J. Sel. Areas Commun. **31**(6), 1084–1094 (2013)
8. Wang, C., Chen, W., Wang, Y.: Scalable influence maximization for independent cascade model in large-scale social networks. Data Min. Knowl. Discov. **25**(3), 545–576 (2012)
9. Yalavarthi, V.K., Khan, A.: Steering top-k influencers in dynamic graphs via local updates. In: IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10–13, 2018. pp. 576–583 (2018)
10. Zhao, T., McAuley, J., King, I.: Improving latent factor models via personalized feature projection for one class recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 821–830. ACM (2015)