Cost-aware Targeted Viral Marketing in Billion-scale Networks

***Abstract*—Online social networks have been one of the most effective platforms for marketing and advertising. Through the "world-of-mouth" exchanges, so-called viral marketing, the influence and product adoption can spread from few key influencers to billions of users in the network. To identify those key influencers, a great amount of work has been devoted for the Influence Maximization (IM) problem that seeks a set of k seed users that maximize the expected influence. Unfortunately, IM encloses two impractical assumptions: 1) any seed user can be acquired with the same cost and 2) all users are equally interested in the advertisement. In this paper, we propose a new problem, called *Cost-aware Targeted Viral Marketing* (CTVM), to find the most cost-effective seed users who can influence the most relevant users to the advertisement. Since CTVM is NP-hard, we design an efficient $(1 - 1/\sqrt{e} - \epsilon)$-approximation algorithm, named BCT, to solve the problem in billion-scale networks. Comparing with IM algorithms, we show that BCT is both theoretically and experimentally faster than the state-of-the-arts while providing better solution quality. Moreover, we prove that under the Linear Threshold model, BCT is the first *sub-linear time* algorithm for CTVM (and IM) in dense networks. In our experiments with a Twitter dataset, containing 1.46 billions of social relations and 106 millions tweets, BCT can identify key influencers in each trending topic in only few minutes.**

I. Introduction

With billions of active users, Online social networks (OSNs) such as Facebook, Twitter and LinkedIn have become critical platforms for marketing and advertising. Through the "word-of-mouth" exchanges, information, innovation, and brand-awareness can disseminate widely over the network. Many notable examples includes the ALS Ice Bucket Challenge, resulting in more than 2.4 million uploaded videos on Facebook and $98.2m donation to the ALS Association in 2014; the customer initiative #PlayItForward of ToyRUs on Twitter that draws more than $35.5m; and the unrest in many Arab countries in 2012. Despite the huge economic and political impact, viral marketing in billion-scale OSNs is still a challenging problem due to the huge numbers of users and social interactions.

A central problem in viral marketing is the *Influence Maximization* (IM) problem that seeks a seed set of k *influential* individuals in a social network that can (directly and indirectly) influence the maximum number of people. Kempe et al. [1] was the first to formulate IM as a combinatorial optimization problem on the two pioneering diffusion models, namely, *Independent Cascade* (IC) and *Linear Threshold* (LT). Since IM is NP-hard, they provide a natural greedy algorithm that yields $(1-1/e-\epsilon)$-approximate solutions for any $\epsilon > 0$. This celebrated work has motivated a vast amount of work on IM in the past decade [2]–[8].

Unfortunately, the formulation of viral marketing as the IM problem encloses two impractical assumptions: 1) any seed user can be acquired with the same cost and 2) the same benefit obtained when influencing one user. The first assumption implies that incentivizing high-profile individuals costs the same as incentivizing common users. This often leads to impractical solutions with unaffordable seed nodes, e.g., the solutions in Twitter often include celebreties like Katy Perry or President Obama. The second assumption can mislead the company to influence "wrong audience" who are neither interested nor potentially profitable. In practice, companies often target not all users but specific sets of potential customers, decided by the factors like age and gender. Moreover, the targeted users can bring different amount of benefit to the company. Thus, simply counting the number of influenced users, as in the case of IM, does not measure the true impact of the campaign and lead to the choosing of wrong seed set. A few recent works attempt to address the above two issues *separately*. In [9] the authors study the *Budgeted Influence Maximization* (BIM) that considers an arbitrary cost for selecting a node and propose an $(1 - 1/\sqrt{e} - \epsilon)$ approximation algorithm for the problem. However, their algorithm is not scalable enough for billion-scale networks. Recently, there is a serial works in [10]–[12] investigating the *Targeted Viral Marketing* (TVM) problem, in which they attempt to influence a subset of users in the network. Unfortunately, all of these methods rely on heuristics strategy and provide no performance guarantees.

In this paper, we introduce the *Cost-aware Targeted Viral Marketing* (CTVM) problem which takes into account both arbitrary cost for selecting a node and arbitrary benefit for influencing a node. Given a social network abstracted by a graph $G = (V, E)$, each node u represents a user with a *cost* $c(u)$ to select into the seed set and a *benefit* $b(u)$ obtained when u is influenced. Given a budget B, the goal is to find a seed set S with total cost at most B that maximizes the expected total benefit over the influenced nodes. CTVM is more relevant in practice as it generalizes other viral marketing problems including TVM, BIM and the fundamental IM. However, the problem is much more challenging with heterogeneous costs and benefits. As we show in Section 3, extending the state-of-the-art method for IM in [8] may increase the running time by a factor $|V|$, making the method unbearable for large networks.

We introduce BCT, an efficient approximation algorithm for CTVM for billion-scale networks. Given arbitrarily small $\epsilon > 0$, our algorithm guarantees a $(1-1/\sqrt{e}-\epsilon)$-approximate solution in general case and a $(1 - 1/e - \epsilon)$-approximate solution when nodes have uniform costs. BCT also outperforms TIM/TIM+, the state-of-the-art methods for IM, when nodes

have uniform costs and benefits. In particular, BCT only takes several minutes to process a network with 41.7 million nodes and 1.5 billion edges. Our contributions are summarized as follows.

- We propose the *Cost-aware Targeted Viral Marketing* (CTVM) problem that consider *heterogeneous costs and benefits* for nodes in the network. Our problem generalizes other viral marketing problems including TVM, CTVM, and the fundamental IM problems.
- We propose BCT, an efficient algorithm that returns $(1-1/\sqrt{e}-\epsilon)$-approximate solutions for CTVM with a high probability. The two novel aspects of BCT are an efficient benefit sampling strategy (Section III) and an efficient stopping rule (Section IV) that guarantees an asymptotic minimal number of samples. Interestingly, the time complexity is independent of the number of edges under the LT model, making BCT the *first sub-linear time algorithm* for CTVM (and IM) in dense graphs.
- We perform extensive experiments on various real networks. BCT, considering both cost and benefit, provides significantly higher quality solutions than existing methods, while running multiple times faster than the state-of-the art ones. Further, we also demonstrate the ability of BCT to identify key influencers in trending topics in a Twitter dataset of 1.5 billion social relations and 106 million tweets within few minutes.

Related works. Kempe et al. [1] is the first to formulate IM as an optimization problem. They show the problem to be NP-complete and devise an $(1-1/e-\epsilon)$ approximation algorithm. Also, IM cannot be approximated within a factor $(1-\frac{1}{e}+\epsilon)$ [13] under a typical complexity assumption. Later, computing the exact influence is shown to be #P-hard [3]. Leskovec et al. [2] study the influence propagation in a different perspective in which they aim to find a set of nodes in networks to detect the spread of virus as soon as possible. They improve the simple greedy method with the lazy-forward heuristic (CELF), which is originally proposed to optimize submodular functions in [14], obtaining an (up to) 700-fold speed up.

Several heuristics are developed to derive solutions in large networks. While those heuristics are often faster in practice, they fail to retain the $(1-1/e-\epsilon)$-approximation guarantee and produce lower quality seed sets. Chen et al. [15] obtain a speed up by using an influence estimation for the IC model. For the LT model, Chen et al. [3] propose to use local directed acyclic graphs (LDAG) to approximate the influence regions of nodes. In a complement direction, there are works on learning the parameters of influence propagation models [16], [17].

Recently, Borgs et al. [18] make a theoretical breakthrough and present an $O(kl^2(m+n)\log^2 n/\epsilon^3)$ time algorithm for IM under IC model. Their algorithm (RIS) returns a $(1-1/e-\epsilon)$-approximate solution with probability at least $1-n^{-l}$. In practice, the proposed algorithm is, however, less than satisfactory due to the rather large hidden constants. In a sequential work, Tang et al. [8] reduce the running time to $O((k+l)(m+n)\log n/\epsilon^2)$ and show that their algorithm is also very efficient in billion-scale networks. However, we show in Section III that the straightforward adaption of the methods in [18] and [8] for CTVM can incur an excessive number of samples, thus, are not efficient enough for large networks.

In another work, Nguyen and Zheng [19] investigate the BIM problem in which each node can have an arbitrary selecting cost. They proposed a $(1-1/\sqrt{e}-\epsilon)$ approximation algorithm (called BIM) based on a greedy algorithm for Budgeted Max-Coverage in [20] and two other heuristics. However, none of the proposed algorithms can handle billion-scale networks.

A line of works in [10]–[12] consider Topic-aware Influence Maximization problem in which edges are associated with a topic-dependent user-to-user social influence strengths. The problem also asks for a set of k users that maximize user adoptions. However, all of the proposed methods do not possess any theoretical guarantess on the solution quality.

Organization. The rest of the paper is organized as follows. In Section II, we present network model, propagation models, and the problem definitions. Section III presents our BCT algorithm for CTVM. We analyze BCT approximation factor and time complexity in Section IV. Experimental results on real social networks are shown in Section V. We conclude in Section VI.

II. Models and Problem Definitions

In this section, we formally define the CTVM problem and present an overview of the Reverse Influence Sampling approaches in Borgs et al. [18] and Tang et al. [8]. For readability, we focus on the *Linear Threshold* (LT) propagation model [1] and summarize our solutions for the *Independent Cascade* (IC) model in Section 4.A.

A. Model and Problem Definition

Let $G=(V,E,c,b,w)$ be a social network with a node set V and a directed edge set E, with $|V|=n$ and $|E|=m$. Each node $u\in V$ has a selecting cost $c(u)\geq 0$ and a benefit $b(u)$ if u is influenced. Each directed edge $(u,v)\in E$ is associated with an influence weight $w(u,v)\in[0,1]$ such that $\sum_{u\in V} w(u,v)\leq 1$. Given G and a subset $S\subset V$, referred to as the *seed set*, in the LT model the influence cascades in G as follows. First, every node $v\in V$ independently selects a *threshold* λ_v uniformly at random in $[0,1]$. Next the influence propagation happens in round $t=1,2,3,\ldots$.

- At round 1, we *activate* nodes in the seed set S and set all other nodes *inactive*. The cost of activating the seed set S is given $c(S)=\sum_{u\in S} c(u)$.
- At round $t>1$, an inactive node v is activated if the weighted number of its activated neighbors reaches its threshold, i.e., $\sum_{\text{active neighbor } u} w(u,v)\geq\lambda_v$.
- Once a node becomes activated, it remains activated in all subsequent rounds. The influence propagation stops when no more nodes can be activated.

Denote by $\mathbb{I}(S)$ the expected number of activated nodes given the seed set S, when the expectation is taken among all λ_v values from their uniform distributions. We call $\mathbb{I}(S)$ the *influence spread* of S in G under the LT model.

The LT is shown in [1] to be equivalent to the reachability in a random graph g, called *live-edge graph* or *sample graph*,

defined as follows: Given a graph $G = (V, E, w)$, for every $v \in V$, select at most one of its incoming edges at random, such that the edge (u, v) is selected with probability $w(u, v)$, and no edge is selected with probability $1 - \sum_u w(u, v)$. The selected edges are called *live* and all other edges are called *blocked*. By claim 2.6 in [1], the influence spread of a seed set S equals the expected number of nodes reachable from S over all possible sample graphs, i.e.,

$$\mathbb{I}(S) = \sum_{g \sqsubseteq G} \Pr[g]|R(g, S)|,$$

where $\sqsubseteq$ denotes that the sample graph g is generated from G with a probability denoted by $\Pr[g]$, and $R(g, S)$ denotes the set of nodes reachable from S in g.

Similarly, the *benefit* of a seed set S is defined as the expected total benefit over all influenced nodes, i.e.,

$$\mathbb{B}(S) = \sum_{g \sqsubseteq G} \Pr[g] \sum_{u \in R(g,S)} b(u).$$

We are now ready to define our problem as follows.

Definition 1 (Cost-aware Targeted Viral Marketing -CTVM)**.** *Given a graph $G = (V, E, c, b, w)$ and a budget $B > 0$, find a seed set $S \subset V$ with total cost $c(S) \leq B$ to maximize the benefit $\mathbb{B}(S)$.*

CTVM generalizes the following viral marketing problems.

- Influence Maximization (IM): IM is a special case of CTVM with $c(u) = 1$ and $b(u) = 1\ \forall u \in V$.
- Budgeted Influence Maximization (BIM) [19]: find a seed set with total cost at most B, that maximizes $\mathbb{I}(S)$. That is $b(u) = 1\ \forall u \in V$.
- Targeted Viral Marketing (TVM): find a set of k node to maximize the number of influenced nodes in a targeted set T. This is $c(u) = 1\ \forall u \in V$ and benefits $c(v) = 1$ if $v \in T$, and $c(w) = 0$ otherwise.

Since IM is a special case of CTVM, CTVM inherits the IM's complexity and hardness of approximation. Thus CTVM is an NP-hard problem and cannot be approximated within a factor $1 - 1/e + \epsilon$ for any $\epsilon > 0$, unless $P = NP$.

In Table I, we summarize the frequently used notations.

B. Summary of the RIS Approach

The major bottle-neck in previous methods for IM [1], [2], [4], [19] is the inefficiency in estimating the influence spread. To address this, Borgs et al. [18] introduced a novel approach for IM, called Reverse Influence Sampling (RIS), which is the foundation for TIM/TIM+ algorithms, the state-of-the-art methods for IM [8].

Given $G = (V, E, w)$, RIS captures the influence landscape of G through generating a hypergraph $\mathcal{H} = (V, \{\mathcal{E}_1, \mathcal{E}_2, \ldots\})$. Each hyperedge $\mathcal{E}_j \in \mathcal{H}$ is a subset of nodes in V and constructed as follows.

Definition 2 (Random Hyperedge)**.** *Given $G = (V, E, w)$, a random hyperedge $\mathcal{E}_j$ is generated from G by 1) selecting a random node $v \in V$ 2) generating a sample graph $g \sqsubseteq G$ and 3) returning $\mathcal{E}_j$ as the set of nodes that can reach v in g.*

TABLE I: Table of Symbols

Notation	Description
n, m	#nodes, #links in G, respectively
$\mathbb{I}(S), \mathbb{I}(S, u)$	Influence Spread of seed set $S \subseteq V$ and influence of S on a node v. For $v \in V$, $\mathbb{I}(v) = \mathbb{I}(\{v\})$
Γ	Sum of all node benefits, $\sum_{v \in V} b(v)$
$\mathbb{B}(S)$	Benefit of seed set $S \subseteq V$
$\hat{\mathbb{B}}(S)$	$\mathbb{B}'(S) = \frac{deg_{\mathcal{H}}(S)}{m_{\mathcal{H}}}\Gamma$ - an estimator of $\mathbb{B}(S)$
OPT_k	The maximum $\mathbb{B}(S)$ for any size-k seed set S
S_k^*	An optimal size-k seed node, i.e., $\mathbb{B}(S_k^*) = OPT_k$
$m_{\mathcal{H}}$	#hyperedges in hypergraph $\mathcal{H}$
$deg_{\mathcal{H}}(S)$, $S \subseteq V$	#hyperedges incident at some node in S. Also, $deg_{\mathcal{H}}(v)$ for $v \in V$
c	$c = 2(e - 2) \approx \sqrt{2}$
Υ_L^u	$\Upsilon_L^u = 8c(1 - \frac{1}{2e})^2 \left[\ln \frac{1}{\delta} + \ln \binom{n}{k} + \frac{2}{n}\right] \frac{1}{\epsilon^2}$ $\leq 3.7 \left[\ln \frac{1}{\delta} + \ln \binom{n}{k} + \frac{2}{n}\right] \frac{1}{\epsilon^2}$
Υ_L^c	$\Upsilon_L^c = 8c(1 - \frac{1}{2e})^2 \left[\ln \frac{1}{\delta} + k_{max} \ln n + \frac{2}{n}\right] \frac{1}{\epsilon^2}$
Λ_L	$\Lambda_L = (1 + \frac{e\epsilon}{2e-1})\Upsilon_L$
M_k	$M_k = \binom{n}{k} + 2$

Node v in the above definition is called the *source* of $\mathcal{E}_j$ and denoted by $\mathsf{src}(\mathcal{E}_j)$. Observe that $\mathcal{E}_j$ contains the nodes that can influence its source v. If we generate multiple random hyperedges, influential nodes will likely appear more often in the hyperedges. Thus a seed set S that *covers* most of the hyperedges will likely maximize the influence spread $\mathbb{I}(S)$. Here a a seed set S covers a hyperedge $\mathcal{E}_j$, if $S \cap \mathcal{E}_j \neq \emptyset$. This observation is captured in the following lemma in [18].

We denote by $m_{\mathcal{H}}$ the number of hyperedges in $\mathcal{H}$.

Lemma 1. *[18] Given $G = (V, E, w)$ and a random hyperedge $\mathcal{E}_j$ generated from G. For each seed set $S \subset V$,*

$$\mathbb{I}(S) = n \Pr[S \text{ covers } \mathcal{E}_j]. \tag{1}$$

RIS framework. Based on the above lemma, the IM problem can be solved using the following framework.

- Generate multiple random hyperedges from G
- Use the greedy algorithm for the Max-coverage problem [20] to find a seed set S that covers the maximum number of hyperedges and return S as the solution.

The core issue in applying the above framework is that: *How many hyperedges are sufficient to provide a good approximation solution?* For any $\epsilon, \delta \in (0, 1)$, Tang et. al. established in [8] a theoretical threshold

$$\theta = (8 + 2\epsilon) n \frac{\ln 2/\delta + \ln \binom{n}{k}}{\epsilon^2 OPT_k^{IM}}, \tag{2}$$

and proved that when the number of hyperedges in $\mathcal{H}$ is at least θ, the above framework returns a $(1 - 1/e - \epsilon)$-approximate solution with probability $1 - \delta$. Here OPT_k^{IM} denotes the maximum influence spread $\mathbb{I}(S)$ among all size-k seed set.

Unfortunately, computing OPT_k^{IM} is intractable, thus, the proposed algorithms TIM/TIM+ in [8] have to generate $\theta \frac{OPT_k^{IM}}{KPT^+}$ hyperedges, where the ratio $\frac{OPT_k^{IM}}{KPT^+} \geq 1$ is not

upper-bounded. That is TIM/TIM+ may generate many times more hyperedges than needed. In contrast, our BCT algorithm in Section IV guarantees that the number of hyperedges is at most a constant time of the theoretical threshold (with a high probability). Thus, its running time is both smaller and more predictable.

C. Difficulty in Extending RIS to Estimate Benefit $\mathbb{B}(S)$

The most intuitive way to extend the RIS framework to cope with benefit of the nodes is to modify the RIS framework to find a seed set S that covers the maximum *weighted* number of hyperedges, where the weight of a hyperedge $\mathcal{E}_j$ is the benefit of the source $\mathsf{src}(\mathcal{E}_j)$.

Given a seed set $S \subset V$, define a random variable $X'_j = b(\mathsf{src}(\mathcal{E}_j)) \times 1_{(S \text{ covers} \mathcal{E}_j)}$, i.e., $X'_j = b(\mathsf{src}(\mathcal{E}_j))$ if $S \cap \mathcal{E}_j \neq \emptyset$ and $X'_j = 0$, otherwise. We can show, similar to the Lem. 1, that

$$\mathbb{B}(S) = n\mathbb{E}[X'_j]$$

Then we can follow the same approach in Tang et al. [8] to establish the theoretical threshold

$$\theta_B = (8 + 2\epsilon)n\mathbf{b_{max}}\frac{\ln 2/\delta + \ln \binom{n}{k}}{\epsilon^2 OPT_k}, \tag{3}$$

where OPT_k is the maximum benefit $\mathbb{B}(S)$ for any size-k seed set S and $b_{max} = \max\{b(u)|u \in V\}$.

Unfortunately, θ_B can be as large as n times θ in the worst-case. To see this, we can (wlog) normalize the node benefit $b(u)$ so that $\sum_{u\in V} b(u) = n$. Then note that b_{max} could be as large as $\sum_{u\in V} b(u) = n$. One of the reason for the large number of samples is that X'_j can obtain any values among $\{b(u)|u \in V\}$ and thus often has a large variance.

As the above way of extending the RIS to solve CTVM does not scale for large networks, new sampling technique is required for solving CTVM.

III. BCT - A Scalable Approximation Algorithm

In this section, we present BCT - a scalable approximation algorithm for CTVM. BCT combines two novel techniques: BSA, a sampling strategy to estimate the benefit and a powerful stopping condition to smartly detect when the sufficient number of hyperedges is reached.

A. BCT - The Main Algorithm

Algorithm 1 BSA - Benefit Sampling Algorithm for LT model

Input: Weighted graph $\mathcal{G} = (V, E, w)$.
Output: A random hyperedge $\mathcal{E}_j \subseteq V$.
1: $\mathcal{E}_j \leftarrow \emptyset$
2: Pick a node u with probability $\frac{b(u)}{\Gamma}$.
3: **Repeat**
4: Add u to $\mathcal{E}_j$
5: Attempt to select an edge (v, u) using live-edge model
6: **if** edge (v, u) is selected **then** Set $u \leftarrow v$.
7: **Until** $(u \in \mathcal{E}_j)$ OR (no edge is selected)
8: Return $\mathcal{E}_j$

BCT algorithm for the CTVM problem is presented in Algorithm 3. The algorithm uses BSA (Algorithm 1), which will be described in details in subsection III-B, to generate hyperedges and Weighted-Max-Coverage (Algorithm 2) to find a candidate seed set $\hat{S}$ following the RIS framework.

Algorithm 2 Weighted-Max-Coverage Algorithm

Input: Hypergraph $\mathcal{H}$ and Budget B.
Output: Seed set S.
1: $S = \emptyset$
2: **while** $\{v \in V \setminus S | c(v) \leq B - c(S)\} \neq \emptyset$ **do**
3: $\hat{v} \leftarrow \arg\max_{\{v\in V|c(v)\leq B-c(S)\}} \frac{deg_{\mathcal{H}}(S\cup\{v\})-deg_{\mathcal{H}}(S)}{c(v)}$
4: Add $\hat{v}$ to S
5: **end while**
6: $u = \arg\max_{\{v\in V|c(v)\leq B\}} deg_{\mathcal{H}}(v)$
7: **if** $deg_{\mathcal{H}}(S) < deg_{\mathcal{H}}(u)$ **then**
8: $S = \{u\}$
9: **return** S

CTIM keeps generating hyperedges until the degree of the seed set selected by Weighted-Max-Coverage exceeds a threshold Λ_L (the stopping condition). The algorithm runs in rounds and up to the i-th round, it generates $2^{i-1}\Lambda_L$ hyperedges. After each round, Weighted-Max-Coverage algorithm is called to select a seed set $\hat{S}$ within the budget B and stop the algorithm if the degree $\hat{S}$ exceeds Λ_L. Otherwise, it continues to generate more hyperedges. In the worst case, BCT generates twice as many as the theoretical number of hyperedges needed in the stopping condition (Lem. 4).

Algorithm 3 BCT Algorithm

Input: Graph $G = (V, E, b, c, w)$, budget $B > 0$, and $\epsilon, \delta \in (0, 1)$.
Output: Seed set S_k.
1: $\Upsilon_L = \Upsilon_L^u$ for uniform cost and $\Upsilon_L = \Upsilon_L^c$ otherwise
2: $\Lambda_L = (1 + \frac{e\epsilon}{2e-1})\Upsilon_L$
3: $N_t = \Lambda_L$
4: $\mathcal{H} \leftarrow (V, \mathcal{E} = \emptyset)$
5: **repeat**
6: **for** $j = 1$ **to** $N_t - |\mathcal{E}|$ **do**
7: Generate $\mathcal{E}_j \leftarrow$BSA($\mathcal{G}$)
8: Add $\mathcal{E}_j$ to $\mathcal{E}$.
9: **end for**
10: $N_t = 2N_t$
11: $\hat{S} =$ Weighted-Max-Coverage($\mathcal{H}$, B)
12: **until** $deg_{\mathcal{H}}(\hat{S}) \geq \Lambda_L$
13: **return** $\hat{S}$

The Weighted-Max-Coverage algorithm is the weighted version of the greedy strategy for Max-Coverage problem presented in [20] to find a maximum cover within the budget B. This procedure considers two candidates and chooses the one with higher coverage: one is taken from greedy strategy (Lines 1-5) and the another is just a node having highest coverage within the budget. In [20], the authors prove that this procedure returns a $(1 - 1/\sqrt{e})$-approximate cover in the general case of arbitrary cost. However, if the node cost is uniform, Weighted-Max-Coverage considers only the candidate obtained by greedy strategy and has the approximation factor of $(1 - 1/e - \epsilon)$.

B. Efficient Benefit Sampling Algorithm - BSA

Due to the inefficiency of RIS when applying to CTVM problem, we propose an efficient adapted version of RIS, called Benefit Sampling Algorithm - BSA, for estimating benefit $\mathbb{B}(S)$. The BSA for generating a random hyperedge $\mathcal{E}_j \subseteq V$ under LT model is summarized in Algorithm 1. The procedure for IC model is similar except for the generating of live-edge in the Line 5. The great deal of difference of BSA from RIS is that it *chooses the source node proportional to benefit of each*

node as opposed to choosing uniformly at random in RIS. That is the probability of choosing node u is $P(u) = b(u)/\Gamma$ with $\Gamma = \sum_{v \in V} b(v)$. After choosing a starting node u, it attempts to select an *in-neighbor* v of u according to the LT model and make (v, u) a *live* edge. Then it "move" to v and repeat the process. The procedure stops when we encounter a previously visited vertex or no edge is selected. The hyperedge is the set of nodes visited along the process.

Note that the selection of a source node with the probability proportional to the benefit can be done in $O(1)$ after an $O(n)$ preprocessing using the Alias method [21]. Similarly, the selection of the live edge according to the influence weight can also be done in $O(1)$. In contrast, in the IC model [18], it takes a time $\theta(d(v))$ at a node v to generate all live edges pointing to v. *This key difference makes the generating hyperedges in the LT model much more efficient than that in the IC model.*

The key insight into why random hyperedges generated via BSA can capture the benefit landscape is stated in the following lemma.

Lemma 2. *Given a fixed seed set $S \subseteq V$, for a random hyperedge $\mathcal{E}$,*

$$\Pr[\mathcal{E}_j \cap S \neq \emptyset] = \frac{\mathbb{B}(S)}{\Gamma}$$

Proof.

$$\begin{aligned}
\mathbb{B}(S) &= \sum_{u \in V} \Pr_{g \sqsubseteq G}[u \in R(g, S)] b(u) \\
&= \sum_{u \in V} \Pr_{g \sqsubseteq G}[\exists v \in S \text{ such that } v \in \mathcal{E}_j(u)] b(u) \\
&= \Gamma \sum_{u \in V} \Pr_{g \sqsubseteq G}[\exists v \in S \text{ such that } v \in \mathcal{E}_j(u)] \frac{b(u)}{\Gamma} \\
&= \Gamma \Pr_{g \sqsubseteq G, u \in V}[\exists v \in S \text{ such that } v \in \mathcal{E}_j] \\
&= \Gamma \Pr_{g \sqsubseteq G, u \in V}[S \cap \mathcal{E}_j \neq \emptyset] \qquad (4)
\end{aligned}$$

Since we select u with probability $P(u) = b(u)/\Gamma$, the forth equality contains the expected probability taken over the benefit distribution.

IV. Approximation and Complexity Analysis

In this section, we prove that BCT returns a $(1-1/e-\epsilon)$-approximate solution for uniform cost version of CTVM problem and a $(1-1/\sqrt{e}-\epsilon)$ solution for the arbitrary cost version. We also analyze the time complexity of BCT and show an interesting result that, for LT model, BCT has sub-linear time complexity.

A. Approximation Guarantee for uniform cost CTVM

In this subsection, we will prove the approximation factor of BCT to be $(1-\frac{1}{e}-\epsilon)$ for uniform cost CTVM problem where all nodes have the same cost. First, we show that BCT generates at least $T_k^* = \frac{n\Upsilon_L}{OPT_k}$ hyperedges with high probability in Lemma 4, i.e., our stopping condition. Secondly, we prove that T_k^* hyperedges are sufficient to guarantee that BCT returns an $(1-1/e-\epsilon)$-approximate solution. Combining these results gives us the approximation guarantee of BCT for uniform cost instances of CTVM in Theorem 1.

To prove Lemma 4, we rely on the following Lemma with the proof presented in the Appendix.

Lemma 3. *Given a size-k set S_k, if the hypergraph has $T_k^* = \frac{n\Upsilon_L}{OPT_k}$ hyperedges, then*

$$\Pr[\mathbb{B}(S_k) \leq \hat{\mathbb{B}}(S_k) - \frac{\epsilon e}{2e-1} OPT_k] \leq \frac{\delta}{M_k} \qquad (5)$$

We now present our stopping condition.

Lemma 4 (Stopping condition)**.** *If there exists a set S with $|S| \leq k$ such that $deg_{\mathcal{H}}(S) \geq \Lambda_L$, then*

$$\Pr[m_{\mathcal{H}} \leq T_k^*] < \frac{\delta}{M_k}. \qquad (6)$$

where $M_k = \binom{n}{k} + 2$.

Let define $X_{S_k} = \min\{|S_k \cap \mathcal{E}_j|, 1\}$ to be a random variable corresponding to set S_k, then,

$$\begin{aligned}
\Pr[m_{\mathcal{H}} \leq T_k^*] &= \Pr\Big[\sum_{j=1}^{m_{\mathcal{H}}} X_{S_k} \leq \sum_{j=1}^{T_k^*} X_{S_k}\Big] \\
&= \Pr\Big[deg_{m_{\mathcal{H}}}(S_k) \leq deg_{T_k^*}(S_k)\Big] \\
&\leq \Pr\Big[(1 + \frac{\epsilon e}{2e-1})\Upsilon_L \leq deg_{T_k^*}(S_k)\Big] \\
&\quad \text{(due to the algorithm's stopping condition)} \\
&= \Pr\Big[(1 + \frac{\epsilon e}{2e-1})\Upsilon_L \frac{\Gamma}{T_k^*} \leq deg_{T_k^*}(S_k)\frac{\Gamma}{T_k^*}\Big] \\
&\leq \Pr\Big[(1 + \frac{\epsilon e}{2e-1})OPT_k \leq \hat{\mathbb{B}}_{T_k^*}(S_k)\Big] \\
&\leq \Pr\Big[\mathbb{B}(S_k) + \frac{\epsilon e}{2e-1} OPT_k \leq \hat{\mathbb{B}}_{T_k^*}(S_k)\Big] \leq \frac{\delta}{M_k} \qquad (7)
\end{aligned}$$

The last inequality is followed from Eq. 5 when using T_k^* hyperedges.

Based on Lemma 4, if we can find a set S such that $deg_{\mathcal{H}}(S) \geq \Lambda_L$, then with very high probability, the number of generated hyperedges is at least T_k^*. Next, we show T^*k hyperedges are sufficient to find a good seed set.

Lemma 5. *If the number of samples (hyperedges) $m_{\mathcal{H}} \geq T_k^*$, BCT returns a seed set $\hat{S}$ with*

$$\Pr[\mathbb{B}(\hat{S}) \leq (1 - 1/e - \epsilon)OPT_k] \leq \frac{\delta(M_k - 1)}{M_k}. \qquad (8)$$

For brevity, the proof is presented in the Appendix.

Lemmas 4 and 5 together prove that if there exists a set S where $|S| \leq k$ such that $deg_{\mathcal{H}}(S) \geq \Lambda_L$, then the greedy algorithm for selecting seed set on the hypergraph $\mathcal{H}$ will return a $(1 - 1/e - \epsilon)$-approximate solution. As a result, the following theorem states the approximation guarantee of BCT.

Theorem 1. *BCT selects a set of k nodes, $\hat{S}_k$, satisfying*

$$\mathbb{B}(\hat{S}_k) \geq (1 - 1/e - \epsilon)OPT_k \qquad (9)$$

with probability at least $1 - \delta$.

Proof. BCT algorithm keeps generating hyperedges until the degree of the seed set returned by Weighted-Max-Coverage, $\hat{S}$, exceeds Λ_L. From Lemma 4, we obtain

$$\Pr[m_{\mathcal{H}} \leq T_k^*] < \frac{\delta}{M_k}. \qquad (10)$$

Assume that $m_{\mathcal{H}} \geq T_k^*$, from Lemma 5, we also have

$$\Pr[\mathbb{B}(\hat{S}) \leq (1 - 1/e - \epsilon)OPT_k] \leq \frac{\delta(M_k - 1)}{M_k}. \qquad (11)$$

Combining Eqs. 10 and 11, we derive the probability $P = \Pr[\mathbb{B}(\hat{S}) \geq (1 - 1/e - \epsilon)OPT_k]$ as follows,

$$\begin{aligned}
P &= 1 - \Pr[\mathbb{B}(\hat{S}) \leq (1 - 1/e - \epsilon)OPT_k] \\
&\geq 1 - \Pr[m_{\mathcal{H}} \leq T_k^*] - \Pr[\mathbb{B}(\hat{S}) \leq (1 - 1/e - \epsilon)OPT_k] \\
&= 1 - \frac{\delta}{M_k} - \frac{\delta(M_k - 1)}{M_k} = 1 - \delta \qquad (12)
\end{aligned}$$

Thus, we complete the proof of Theorem 1

B. Time Complexity

We will analyze the time complexity of generating hyperedges and finding seed set by Weighted-Max-Coverage. At the end, we show that BCT has the overall time complexity of $O((\ln \frac{2}{\delta} + \ln M_k)\epsilon^{-2}n)$, noting that $\ln M_k \leq \ln \binom{n}{k} + 2/n$.

Generating hyperedges. Let $v^* = \arg\max_{v \in V} \mathbb{B}(v)$, we define $Y_j = |\{v^*\} \cap \mathcal{E}_j\}|$, a random variable with mean $\mu_Y = \mathbb{B}(v^*)/\Gamma$. In the Appendix, we prove that the expected number of hyperedges is at most Λ/μ_Y and the expected number of edges visited by BSA is at most $\frac{m}{\Gamma}\mathbb{B}(v^*)$. We bound the time complexity of generating hyperedges by the number of edges examined by the following lemma.

Lemma 6. *The expected number of edges examined by BCT for uniform cost CTVM problem is at most*

$$3.7(\ln(1/\delta) + \ln M_k)\epsilon^{-2}n \tag{13}$$

For completeness, the proof is presented in the Appendix.

Time to Find Max-Coverage. Since we double the number of hyperedges in each round of our algorithm, the overall time complexity of finding Max-Coverage is at most twice that of the last run. Furthermore, the procedure to find Max-Coverage in BCT can be implemented in linear-time in terms of the total size of the hyperedges which is bounded by the number of edges examined. Thus, the complexity of finding Max-Coverage is $O((\ln(1/\delta) + \ln M_k)\epsilon^{-2}n)$.

Theorem 2. *BCT has an expected running time for uniform cost CTVM problem*

$$O((\ln(1/\delta) + \ln M_k)\epsilon^{-2}n) \tag{14}$$

The theorem follows from the fact that both generating hyperedges and finding Max-Coverate have the same complexity of $O((\ln(1/\delta) + \ln M_k)\epsilon^{-2}n)$.

Under the LT model, the time complexity does not depend on the number of edges in the original graph, hence, uniform-cost BCT has a sub-linear time complexity in dense graphs.

C. Approximation Algorithm for the arbitrary cost CTVM

With heterogeneous selecting cost, seed sets may have different size. However, we can find a value $k_{max} = \max\{k : \exists S \subset V, |S| = k, c(S) \leq B\}$ by iteratively selecting the smallest cost nodes until reaching the budget B. We then guarantee that all subsets of size up to k_{max} are well approximated. The number of such seed sets are $\sum_{k \leq k_{max}} \binom{n}{k} \leq n^{k_{max}}$. Thus, the required degree in BCT for heterogeneous selecting cost is $\Upsilon_L^u = 8c(1 - \frac{1}{2e})^2[\ln(1/\delta) + k_{max} \ln n + 2/n]\frac{1}{\epsilon^2}$.

In addition, the Weighted-Max-Coverage algorithm used in CTVM only guarantees $(1 - 1/\sqrt{e})$ approximate solutions, as shown in [20]. Putting these modifications together, we have the following Theorem 3. The proof is similar to that of Theorem 1 and is omitted due to the space constraint.

Theorem 3. *Given a budget B, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, BCT for arbitrary cost CTVM problem returns a solution $\hat{S}$,*

$$\mathbb{B}(\hat{S}) \geq (1 - 1/\sqrt{e} - \epsilon)OPT \tag{15}$$

with probability at least $1 - \delta$ and runs in time

$$O((\ln(1/\delta) + k_{max} \ln n + 2/n)\epsilon^{-2}n) \tag{16}$$

D. Extension to IC model

When applying BCT for IC model, the only change is in the BSA procedure to generate hyperedges following the IC model, as originally presented in [18]. We have the following Theorems for the performance of BCT under the IC model.

Theorem 4. *Given a budget B, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, BCT for uniform cost CTVM problem returns a solution $\hat{S}$,*

$$\mathbb{B}(\hat{S}) \geq (1 - 1/e - \epsilon)OPT \tag{17}$$

with probability at least $1 - \delta$ and runs in time

$$O((\ln(1/\delta) + \ln M_k)\epsilon^{-2}(m + n)) \tag{18}$$

Theorem 5. *Given a budget B, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, BCT for arbitrary cost CTVM problem returns a solution $\hat{S}$,*

$$\mathbb{B}(\hat{S}) \geq (1 - 1/\sqrt{e} - \epsilon)OPT \tag{19}$$

with probability at least $1 - \delta$ and runs in time

$$O((\ln(1/\delta) + k_{max} \ln n + 2/n)\epsilon^{-2}(m + n)) \tag{20}$$

The proofs are omitted to save space.

V. Experiments

In this section, we evaluate and compare the performance of BCT to other influence maximization methods on three aspects: *the solution quality, the scalability, and the applicability* of BCT on a billion-scale dataset with both links and content.

TABLE II: Datasets' Statistics

Dataset	#Nodes	#Edges	Type	Avg. degree
NetHELP [3]	15K	59K	undirected	4.1
NetPHY [3]	37K	181K	undirected	13.4
Enron [22]	37K	184K	undirected	5.0
Epinions [3]	132K	841K	directed	13.4
DBLP [3]	655K	2M	undirected	6.1
Twitter [23]	41.7M	1.5G	directed	70.5

A. Experimental Settings

All the experiments are run on a Linux machine with 2.2Ghz Xeon 8 core processor and 64GB of RAM.

Algorithms compared. We choose three groups of methods to test on: 1) designed for IM task, including the top four state-of-the-art algorithms, i.e., TIM, TIM+ [8], CELF++ [5] and SIMPATH [4]; 2) designed for BIM task, namely, BIM [19] and 3) our method BCT. In the first experiment, we will compare between groups with CTVM problem and the second experiment report results on IM individually.

Datasets. For experimental purpose, we choose a set of 6 datasets from various disciplines: NetHEPT, NetPHY, DBLP are citation networks, Email-Enron is communication network, Twitter and Epinions are online social networks. The description summary of those datasets is in Table II.

Parameter Settings. *Computing the edge weights*. Following the conventional computation as in [4], [8], [19], [24], the weight of the edge (u, v) is calculated as $w(u, v) = \frac{1}{d_{in}(v)}$ where $d_{in}(v)$ denotes the in-degree of node v.

Computing the node costs. Intuitively, the more famous one is, the more difficult it is to convince that person. Hence, we assign the cost of a node proportional to the out-degree of that node: $c(u) = \frac{nd^o(u)}{\sum_{v \in V}(d^o(v))}$.

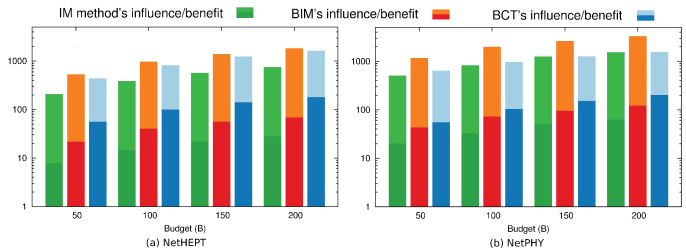


Fig. 1: Comparison between different methods on CTVM task with various budget limits. The whole column indicates influence of the selected seeds while the dark color portion of each column is the benefit of that seed set.

Computing the node benefits. In the first experiment, we choose a random $p = 20\%$ of all the nodes to be the target set and assign benefit 1 to all of them while in case studies, the benefit is learned from a separate dataset.

In all the experiments, we keep $\epsilon = 0.1$ and $\delta = 1/n$ as a general setting or directly stated otherwise. For the other parameters, we take the recommended values in the corresponding papers if available.

B. Experimental results

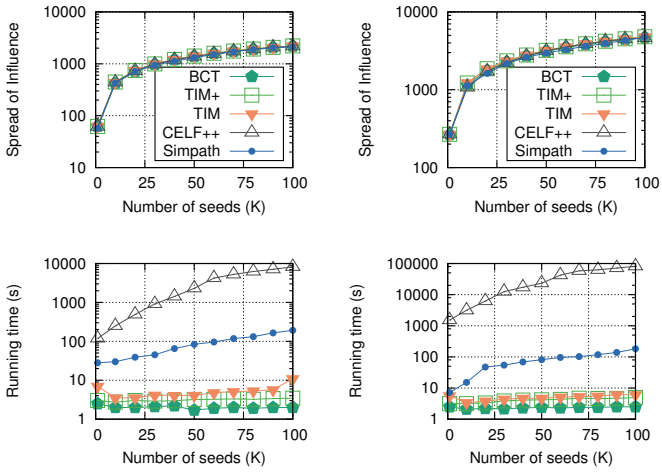
We carry three experiments on both CTVM and IM tasks to compare the performance of BCT with other state-of-the-art methods. In the first experiments, we compare three groups of algorithms, namely, IM based methods, BIM and BCT on CTVM problem. We choose four algorithms in the category of IM methods: CELF++, SIMPATH, TIM and TIM+, which are well known algorithms for IM. The results are presented in Fig. 1. We conduct the second and third experiments on the classical IM task with different datasets and various k values. The results are shown in Table III and Fig. 2.

TABLE III: Comparison between different methods on IM task and various datasets (with $\epsilon = 0.1, k = 50, \delta = \frac{1}{n}$).

Method	Spread of Influence			Running Time (s)		
	Epin.	*Enron*	*DBLP*	*Epin.*	*Enron*	*DBLP*
BCT	**16320**	**16776**	**108600**	**3**	**2**	**5**
TIM+	16293	16732	108343	6	3	12
TIM	16306	16749	107807	8	4	17
Simpath	16291	16729	103331	23	18	136

1) Comparison of solution quality: From Fig. 1, we can see that BCT outperforms the other methods by a large margin on CTVM problem. With the same amount of budget, CTVM returns a solution which is in order of magnitudes better than that of BIM and IM based methods. Because IM algorithms only desire to maximize the influence or aim at the influential nodes, unfortunately, those are usually very expensive nodes. As a consequence, when nodes have arbitrary cost, IM methods suffer severely in terms of both influence and benefit. On the other hand, BIM optimizes cost and influence while ignoring benefit of influencing nodes that causes BIM to select cheaper nodes with high influence. Consequently, the seed sets returned by BIM have high influence but low benefit. On IM task, as shown in Table III and Fig. 2, BCT even marginally surpasses the state-of-the-art methods for IM.

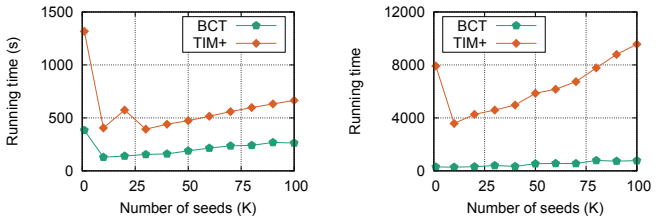
2) Comparison of running time: The experimental results in Table III and Fig. 2 confirms our theoretical establishment



(a) NetHEP (b) NetPHY

Fig. 2: Comparison between different methods on LT model

in Section IV that BCT for uniform cost CTVM requires less than half of the number of hyperedges needed by TIM and TIM+. As such, the running time of BCT in all the experiments are significantly lower than the other methods. In average, BCT is twice as fast as TIM+, up to four times faster than TIM. Since both Simpath and CELF++ require intensive graph simulation, these methods have very poor performance compared to BCT, TIM and TIM+ which apply advanced techniques to approximate the influence landscape.



(a) LT model. (b) IC model.

Fig. 3: BCT and TIM+ on Twitter

C. Twitter: A billion-scale social network

In this subsection, we design two case studies on Twitter network: one is to compare the scalability of BCT with TIM+ - the fastest existing method and the another is using BCT to find a set of users who have highest benefit with respect to a particular topic in Twitter.

1) Compare BCT versus TIM+: Figure 3 shows the results of running BCT and TIM+ on Twitter network dataset using both LT and IC models with k ranging from 1 to 100. Twitter has 1.5 billion edges and all the other methods, except BCT and TIM+, fail to process it within a day in our experiments. The results, here, are consistent with the other results in the previous experiments. Regardless of the values of k, in LT model, BCT is up to 3 times faster than TIM+ and in IC model, this ratio is in order of magnitude since influence in IC model is much larger an, thus, harder for TIM+ to estimate.

We also measure the memory consumed by these two algorithms and observe that, in average, BCT requires around 20GB but TIM+ always need more than 30GB. This is a reasonable results since in addition the memory for the original graph, BCT needs half of the hyperedges generated by TIM+.

2) A Case Study on Twitter network.: We choose two most popular topics with related keywords as reported in [23]. Based on the list of keywords, we use a Twitter's tweet dataset to extract a list of users who mentioned the keywords in their posts and the number of those posts. The number of posts reveals the interest of the users on the topic, thus, we consider this as the benefit of the nodes and run BCT.

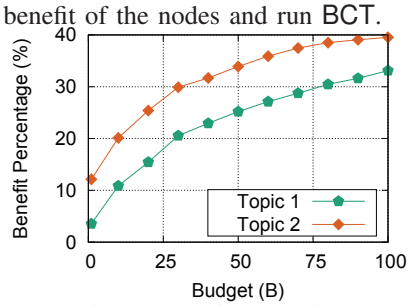


Fig. 4: Benefit on Twitter.

Figure 4 shows the benefit with different budgets for the two topics. We see that apparently the very first chosen nodes have high benefit and it continues increasing later but with much lower rate. Table IV represents the topic, keywords and the users selected by BCT. Looking into the first 5 Twitters chosen by the algorithm, they are users with only few thousands of followers (unlike Katy Perry or President Obama who got more than 50 millions followers) but are highly active poster in the corresponding topic. For example, the first selected users is a Canadian poster, who has about 4000 followers and but generate about 210K posts on the movements of governments in the US and Iran.

VI. Conclusion

In this paper, we propose the CTVM problem that generalizes several viral marketing problems including the classical IM. We propose BCT an efficient approximation algorithm to solve CTVM in billion-scale networks and show that it is both theoretically sound and practical for large networks. The algorithm can be employed to discover more practical solutions for viral marketing problems, as illustrated through the discovering of influential users w.r.t. trending topics in Twitter media site.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD'03*. ACM New York, NY, USA, 2003, pp. 137–146.

[2] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *ACM KDD '07*. New York, NY, USA: ACM, 2007, pp. 420–429.

[3] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM KDD '10*. New York, NY, USA: ACM, 2010, pp. 1029–1038.

[4] A. Goyal, W. Lu, and L. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 211–220.

[5] ——, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*. ACM, 2011, pp. 47–48.

[6] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 629–638.

[7] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-i. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[8] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 75–86.

[9] N. P. Nguyen, G. Yan, and M. T. Thai, "Analysis of misinformation containment in online social networks," *Comput. Netw.*, vol. 57, no. 10, pp. 2133–2146, Jul. 2013.

[10] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," *Knowledge and information systems*, vol. 37, no. 3, pp. 555–584, 2013.

[11] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries." in *EDBT*, 2014, pp. 295–306.

[12] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang, "Online topic-aware influence maximization," *Proceedings of the VLDB Endowment*, vol. 8, no. 6, pp. 666–677, 2015.

[13] U. Feige, "A threshold of ln n for approximating set cover," *Journal of ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[14] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, ser. Lecture Notes in Control and Information Sciences, J. Stoer, Ed. Springer, 1978, vol. 7, pp. 234–243.

[15] N. Chen, "On the approximability of influence in social networks," *SIAM Journal of Discrete Mathematics*, vol. 23, no. 3, pp. 1400–1415, 2009.

[16] A. Goyal, F. Bonchi, and L. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 241–250.

[17] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, "Strip: stream learning of influence probabilities," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 275–283.

[18] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '14. SIAM, 2014, pp. 946–957.

[19] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 6, pp. 1084–1094, 2013.

[20] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[21] A. J. Walker, "An efficient method for generating discrete random variables with general distributions," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 253–256, Sep. 1977. [Online]. Available: http://doi.acm.org/10.1145/355744.355749

[22] B. Klimt and Y. Yang, "Introducing the Enron corpus," in *First Conference on Email and Anti-Spam (CEAS)*, 2004.

[23] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 591–600.

[24] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD '09*. New York, NY, USA: ACM, 2009, pp. 199–208.

[25] A. Wald, *Sequential Analysis*. John Wiley and Sons, 1947.

Appendix

We will use the following lemmas in

Lemma 7. *Let $X_1, ..., X_T$ be i.i.d. random variables with $\mu_{X_i} = \mu$. For any fixed $T > 0$,*

$$\Pr[\hat{\mu} \geq (1+\epsilon)\mu] \leq e^{\frac{-T\mu\epsilon^2}{2c}}$$

and

$$\Pr[\hat{\mu} \leq (1-\epsilon)\mu] \leq e^{\frac{-T\mu\epsilon^2}{2c}}.$$

where $\hat{\mu} = \frac{\sum_{i=1}^{T} X_i}{T}$.

Lemma 8. *Given $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, if we have*

$$T = 2c\ln(2/\delta)\frac{1}{\epsilon^2\mu} \tag{21}$$

i.i.d. random variables $X_1, ..., X_T$ with $\mu_{X_i} = \mu$ then

TABLE IV: Topics, related keywords and first 5 users selected by BCT

Topic	Keywords	#Users	First 5 selected Twitters
1	**bill clinton, iran, north korea, president obama, obama**	997K	dominiquerdr, stockmarketcash, uncoolbobby, larsthebear, dadashiii
2	**senator ted kenedy, oprah, kayne west, marvel, jackass**	507K	royasmusic, bksmarvelous1, edithayala, capitarecesion, dietmission

$$\Pr[|\hat{\mu}-\mu| \geq \epsilon\mu] \geq 1-\delta \quad (22)$$

Proof of Theorem 3 First, we can easily verify that 5 holds with $S_k = S_k^*$ by Lemma 8. For a random set S_k, the inequality 5 is equivalent to

$$\begin{aligned} &\Pr[\mu_{S_k} \leq \hat{\mu}_{S_k} - \frac{\epsilon e}{2e-1}\mu_{S_k^*}] \leq \frac{\delta}{M_k} \\ \Leftrightarrow\ &\Pr[\mu_{S_k} \leq \hat{\mu}_{S_k} - \frac{\epsilon e}{2e-1}\frac{\mu_{S_k^*}}{\mu_{S_k}}\mu_{S_k}] \leq \frac{\delta}{M_k} \end{aligned} \quad (23)$$

Apply 21 to 23, we obtain the number of necessary samples

$$\begin{aligned} T_{S_k} &= 8c(1-\frac{1}{2e})^2\Big[\ln\frac{1}{\delta} + \ln\Big(M_k\Big)\Big]\frac{1}{\epsilon^2\mu_{S_k^*}}\frac{\mu_{S_k}}{\mu_{S_k^*}} \\ &\leq 8c(1-\frac{1}{2e})^2\Big[\ln\frac{1}{\delta} + \ln M_k\Big]\frac{1}{\epsilon^2\mu_{S_k^*}} = T_k^* \end{aligned}$$

since $\mu_{S_k} \leq \mu_{S_k^*}$.

Proof of Theorem 5. To prove Theorem 5, we first need to prove the following inequality when $m_{\mathcal{H}} \geq \frac{n\Upsilon_L}{OPT_k}$

$$\Pr[\hat{\mathbb{B}}(S_k^*) \leq OPT_k - \frac{\epsilon e}{2e-1}OPT_k] \leq \frac{\delta}{M_k} \quad (24)$$

for the optimal solution S_k^*. The left hand side of inequality 24 is equivalent to

$$\begin{aligned} \Pr[\hat{\mu}_{S_k^*} \leq \mu_{S_k^*}(1-\frac{\epsilon e}{2e-1})] &\leq e^{-\frac{T_k^*\mu_{S_k^*}\epsilon^2 e^2}{8c(2e-1)^2}} \text{ (Lem. 7)} \\ &= e^{-\frac{\Upsilon_L\epsilon^2 e^2}{8c(2e-1)^2}} \leq \frac{\delta}{M_k} \end{aligned} \quad (25)$$

The last step is obtained by substituting Υ_L with the definition.

Combining Eqs. 5, 24 and applying union bound over all possible sets of size k and the optimal solution, we have

$$\begin{aligned} &\Pr\Big[\Big(\mathbb{B}(S_k) \leq \hat{\mathbb{B}}(S_k) - \frac{\epsilon e}{2e-1}OPT_k \text{ for all } S_k\Big) \\ &\quad \text{and } \Big(\hat{\mathbb{B}}(S_k^*) \leq OPT_k - \frac{\epsilon e}{2e-1}OPT_k\Big)\Big] \\ &\leq \sum_{S_k}\Pr[\mathbb{B}(S_k) \leq \hat{\mathbb{B}}(S_k) - \frac{\epsilon e}{2e-1}OPT_k] \\ &\quad + \Pr[\hat{\mathbb{B}}(S_k^*) \leq OPT_k - \frac{\epsilon e}{2e-1}OPT_k] \\ &= \frac{\delta}{M_k}\binom{n}{k} + \frac{\delta}{M_k} = \frac{\delta(M_k-1)}{M_K} \end{aligned} \quad (26)$$

In other words, with probability at least $1 - \frac{\delta(M_k-1)}{M_K}$, BCT achieves the followings

$$\begin{aligned} &\mathbb{B}(S_k) \geq \hat{\mathbb{B}}(S_k) - \frac{\epsilon e}{2e-1}OPT_k \\ &\text{and } \hat{\mathbb{B}}(S_k^*) \geq OPT_k - \frac{\epsilon e}{2e-1}OPT_k \end{aligned} \quad (27)$$

Since Weighted-Max-Coverage (Algo. 2) returns $\hat{S}_k$ with $deg_{\mathcal{H}}(\hat{S}_k) \geq (1-1/e)deg_{\mathcal{H}}(S_{k_{max}}) \geq (1-1/e)\ deg_{\mathcal{H}}(S_k^*)$, where $S_{k_{max}}$ is the optimal solution of Weighted-Max-Coverage [13].

Based on 27 and the upper note, we have

$$\begin{aligned} \mathbb{B}(\hat{S}_k) &\geq \hat{\mathbb{B}}(\hat{S}_k) - \frac{\epsilon e}{2e-1}OPT_k \\ &= \frac{\deg_{\mathcal{H}}(\hat{S}_k)}{m_{\mathcal{H}}}\Gamma - \frac{\epsilon e}{2e-1}OPT_k \\ &\geq (1-1/e)\frac{deg_{\mathcal{H}}(S_k^*)}{m_{\mathcal{H}}}\Gamma - \frac{\epsilon e}{2e-1}OPT_k \\ &= (1-1/e)\hat{\mathbb{B}}(S_k^*) - \frac{\epsilon e}{2e-1}OPT_k \\ &\geq (1-1/e)(1-\frac{\epsilon e}{2e-1})OPT_k - \frac{\epsilon e}{2e-1}OPT_k \\ &= (1-1/e-\epsilon)OPT_k \end{aligned} \quad (28)$$

The last step follows from Eq. 5 when we use T_k^* samples.

Proof of Lem. 6 The proof consists of two parts 1) bound the expected number of hyperedges $m_{\mathcal{H}}$ and 2) estimate the mean number of edges visited per reverse influence sampling.

Number of hyperedges: Denote by $\hat{T}(\Lambda_L)$ and $T^*(\Lambda_L)$ the random variables that correspond to the numbers of sampled hyperedges until $deg_{\mathcal{H}}(\hat{S}_k) = \Lambda_L$ and $deg_{\mathcal{H}}(v^*) = \Lambda_L$, respectively. Clearly, $\hat{T}(\Lambda_L) = m_{\mathcal{H}} \leq T^*(\Lambda_L)$, hence,

$$\mathbb{E}[\hat{T}(\Lambda_L)] \leq \mathbb{E}[T^*(\Lambda_L)].$$

Using Wald's equation [25], and that $\mathbb{E}[T^*(\Lambda_L)] < \infty$,

$$\mathbb{E}[T^*(\Lambda_L)]\mu_Y = \Lambda_L$$

Therefore,

$$\mathbb{E}[m_{\mathcal{H}}] = \mathbb{E}[\hat{T}(\Lambda_L)] \leq \mathbb{E}[T^*(\Lambda_L)] = \frac{\Lambda_L}{\mu_Y}.$$

Average number of edges visited per BSA call: The sampling procedure picks a source vertex u proportional to its' benefit. Then for each vertex v, it will choose at most one of the in-neighbors v with a probability $\mathbb{I}(v,u)$, the probability that v can reach to u over all sample graphs of $\mathcal{G}$ (aka the probability that v influences u). Thus the mean number of edges examined by the procedure is

$$\begin{aligned} &\sum_{u\in V}\frac{b(u)}{\Gamma}(\sum_{v\in V}\mathbb{I}(v,u)) = \frac{1}{\Gamma}\sum_{v\in V}\sum_{u\in V}\mathbb{I}(v,u)b(u) \\ &= \frac{1}{\Gamma}\sum_{v\in V}\mathbb{B}(v) \leq \frac{1}{\Gamma}\sum_{v\in V}\mathbb{B}(v^*) = \frac{n}{\Gamma}\mathbb{B}(v^*) \end{aligned} \quad (29)$$

Thus, the expected number of edges visited by BCT is at most

$$\begin{aligned} \frac{n}{\Gamma}\mathbb{B}(v^*)\frac{\Lambda_L}{\mu_Y} &= n\mu_Y\frac{\Lambda_L}{\mu_Y} = n\Lambda_L \\ &\leq 3.7\Big(\ln\frac{1}{\delta} + \ln M_k\Big)\epsilon^{-2}n = O(\Lambda_L n) \end{aligned} \quad (30)$$

This yields the proof.