

Influence Maximization on Social Graphs: A Survey

Yuchen Li, Ju Fan*, Yanhao Wang, and Kian-Lee Tan

Abstract—Influence Maximization (IM), which selects a set of k users (called seed set) from a social network to maximize the expected number of influenced users (called influence spread), is a key algorithmic problem in social influence analysis. Due to its immense application potential and enormous technical challenges, IM has been extensively studied in the past decade. In this paper, we survey and synthesize a wide spectrum of existing studies on IM from an *algorithmic perspective*, with a special focus on the following key aspects: (1) a review of well-accepted diffusion models that capture information diffusion process and build the foundation of the IM problem, (2) a fine-grained taxonomy to classify existing IM algorithms based on their design objectives, (3) a rigorous theoretical comparison of existing IM algorithms, and (4) a comprehensive study on the applications of IM techniques in combining with novel context features of social networks such as topic, location, and time. Based on this analysis, we then outline the key challenges and research directions to expand the boundary of IM research.

Index Terms—Influence Maximization, Information Diffusion, Social Networks, Algorithm Design



1 INTRODUCTION

THE last decades have witnessed the booming of online social networks where hundreds of millions of people interact with each other and produce an unprecedented amount of content. The prevalence of online social networks has prompted much attention on *information diffusion*, as a piece of information could quickly become pervasive through the “word-of-mouth” propagation among friends in the network. This diffusion phenomenon has been shown to be powerful in many applications [40], such as adoption of political standpoints and technical innovations. A very recent example is Donald Trump’s presidential campaign in 2016, where Twitter is almost daily used as a campaign tool. As such, information diffusion in online social networks has attracted extensive research efforts from multiple fields, including computer science, physics, epidemiology, etc.

As a key algorithmic problem in information diffusion research, *influence maximization* (IM) has been extensively studied recently due to its potential commercial value. IM aims to select a set of k users in an online social network, aka. *seed set* with the maximum *influence spread*, i.e., the expected number of influenced users through the seed set in information diffusion is maximized. A well-known application of IM is viral marketing [25], where a company may wish to spread the adoption of a new product from some initially selected adopters through the social links between users. Besides viral marketing, IM is also the cornerstone in many other important applications such as network monitoring [62], rumor control [8], [45], and social recommendation [112].

Despite its immense application potential, IM embraces enormous research challenges. The first challenge is how to model the information diffusion process in a social network, which would heavily affect the influence spread of any seed set in IM. Second, the IM problem is theoretically complex in general. It has been proven that obtaining an optimal solution of IM is NP-hard under most of the diffusion models [10], [50], [71]. Furthermore, due to the stochastic nature of information diffusion, even the evaluation of influence spread of any individual seed set is computationally complex. These theoretical results have shown that it is very challenging to retrieve a (near) optimal seed set and to scale to massive social graphs at the same time. Third, very recently, online social networks are being equipped with novel features, e.g., location-based services, topical analysis, streaming content, etc. This has opened up an opportunity of combining IM with various *contexts*, such as location, time and topic information, in order to improve the effectiveness of IM. Many technical challenges naturally arise in solving such context-aware influence maximization problems.

The aforementioned challenges have driven a proliferation of researches in the past decade on developing techniques for influence maximization. In this paper, we aim to provide a comprehensive survey on IM from an *algorithmic perspective*, and focus on the following three aspects as illustrated in Figure 1.

- Y. Li is with the School of Information Systems, Singapore Management University. Email: yuchenli@smu.edu.sg
- J. Fan is with the Key Lab of Data Engineering and Knowledge Engineering, MOE of China, and the School of Information, Renmin University of China. Email: fanj@ruc.edu.cn
- Y. Wang and K. Tan are with the School of Computing, National University of Singapore. Email: {yanhao90, tankl}@comp.nus.edu.sg
- * Ju Fan is the corresponding author.

• **Problem (Section 2).** The IM problem is defined on *diffusion models* to capture the information diffusion process among the users in an online social network. We thus review several classical and well-accepted diffusion models to build the foundation of the IM problem. Subsequently, we formally define the IM problem and discuss the characteristics as well as the computational complexity of the problem under diffusion models.

• **Algorithm (Sections 3-6).** As IM is NP-hard, existing works focus on approximate solutions, and a keystone of

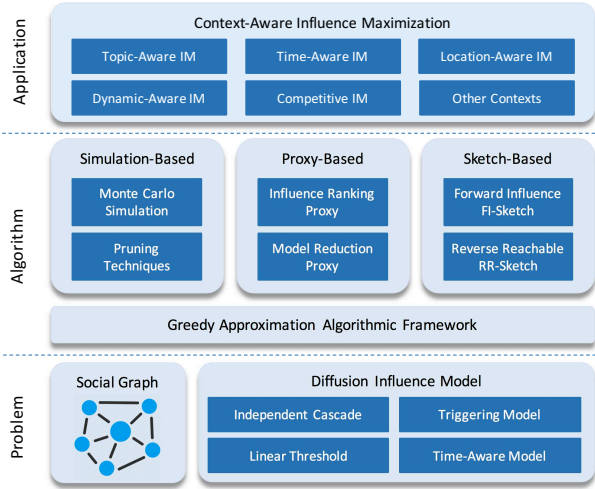


Fig. 1. The survey's overview.

these algorithmic IM studies is the greedy framework. We review the greedy framework and propose a taxonomy to classify existing IM algorithms into the *simulation-based*, the *proxy-based* and the *sketch-based* approaches, based on their algorithmic design for achieving different desired objectives (Section 3). We thoroughly review existing IM algorithms in each approach using a fine-grained classification respectively (Sections 4-6), and provide a rigorous theoretical analysis for comparing theoretical bounds and complexities of the algorithms.

- **Application (Sections 7).** The context-aware IM problems are emerging in recent years. Extended from the classical IM problem, context-aware IM problems consider contextual features such as topical, temporal and spatial information. This survey also reviews the context-aware IM algorithms for two reasons. First, it analyzes how the aforementioned IM approaches are applied as building blocks by combining with contextual information. Second, it introduces how the context-aware features are integrated into the classical IM problem for supporting novel applications. These are prerequisites for developing new algorithms to expand the boundary of the influence maximization research.

Differences from existing surveys. Although there are existing surveys [2], [11], [40], [96], [101], [113] on social influence analysis, this survey is distinct in the following aspects. The authors of [96] focus on diffusion models as well as the methods to train these models from social graph structures or user generated data. Guille et al. [40] narrate a high-level analysis on how existing social influence analysis methods can benefit a broader range of domains in social network studies. Although some surveys like [11], [101], [113] focus on algorithmic methods for IM, they are rather incomplete as an abundance of innovative methods (e.g., sketch-based algorithms) for IM have been developed in recent years (e.g., [30], [84], [99], [100], [107]). Akhil et al. [2] report a comprehensive experimental study on a number of recent IM methods but does not provide any theoretical analysis. Moreover, its experimental design is controversial as pointed out by [75]. Compared with the existing surveys, we focus on presenting a comprehensive review on the state-of-the-art algorithmic methods for IM. We introduce a fine-grained classification and a rigorous theoretical analysis of existing IM algorithms. Moreover, this paper is the first

survey, to the best of our knowledge, on reviewing the recent efforts on context-aware IM and pinpointing their relationships to classical IM algorithms.

To summarize, this paper presents an extensive survey of IM algorithms with the following contributions.

- We develop a fine-grained taxonomy for classifying existing IM algorithms based on their design objectives.
- We present a rigorous theoretical comparative study of the existing algorithms, with a special focus on theoretical bounds and complexity analysis.
- We survey the context-aware IM problems and discuss how IM algorithms are applied as building blocks in designing efficient context-aware IM algorithms.

2 THE INFLUENCE MAXIMIZATION PROBLEM

Influence Maximization (IM) is first modeled as an algorithmic problem by Kempe et al. [50] in 2003. This problem studies a social network represented as a graph $G = (V, E)$, where V is the set of nodes in G (i.e., *users*) and E is the set of (directed/undirected) edges in G (i.e., *social links* between *users*). The goal of the IM problem is to find a k -sized set of users with the maximum *influence* in graph G .

The influence of any seed set is defined based on the *information diffusion* process among the users. An example of the information diffusion is viral marketing, where a company may wish to spread the adoption of a new product from some initial adopters through the social links between users. To quantify information diffusion, we formally define the *diffusion model* and the *influence spread* under the model.

Definition 1 (Diffusion Model & Influence Spread). Given a social graph $G = (V, E)$, a user set $S \subseteq V$, and a diffusion model M captures the stochastic process for S spreading information on G . The influence spread (aka. the *influence function*) of S , denoted as $\sigma_{G,M}(S)$, is the *expected number* of users influenced by S (e.g., users who adopt the new product in viral marketing), where $\sigma_{G,M}(\cdot)$ is a non-negative set function defined on any subset of users, i.e., $\sigma_{G,M}: 2^V \rightarrow \mathbb{R}_{\geq 0}$.

Based on the formalization of the influence spread, the influence maximization problem is defined as follows:

Definition 2 (Influence Maximization (IM) [50]). Given a social graph G , a diffusion model M and a positive integer k , IM selects a set S^* of k users from V as the seed set to maximize the influence spread $\sigma_{G,M}(S^*)$, i.e., $\sigma_{G,M}(S^*) = \arg \max_{S \subseteq V, |S| \leq k} \sigma_{G,M}(S)$. For the ease of presentation, we omit the subscript of $\sigma_{G,M}(\cdot)$ when the context is clear.

Intuitively, the influence function $\sigma(\cdot)$ would heavily depend on the diffusion process. Recent years have witnessed a large amount of literature that develops diffusion models to formulate the diffusion process and compute the influence spread. We will review some commonly-used models in Section 2.1. Moreover, we will also illustrate the computational hardness of IM in Section 2.2.

2.1 Diffusion Models

Recently, there exists a huge amount of literature on designing diffusion models in the areas of data mining, databases, networks, and epidemiology. As the focus of this survey is to review the *algorithmic aspects* of IM, this section reviews the models that are commonly used for IM.

We first present a generic diffusion framework of the reviewed diffusion models. The framework associates each user $u \in V$ with a status of either *inactive* or *active*. Then, based on the social graph G , it considers the following diffusion process among users. Initially, it views the status of a set of chosen users, called *seed set* $S \subseteq V$, to be active, while other users in V are inactive. Then, it considers the diffusion process that the seed users in S can “influence” their neighbors to be active, the newly activated users can further activate their neighbors, and so on. This diffusion process terminates when no new users can be activated. In particular, the framework models the aforementioned “activation” as a stochastic process, the influence spread $\sigma(S)$ is then naturally defined as the expected number of users with *active* status after the diffusion process terminates. In this survey, we focus on *progressive* diffusion models, i.e., activated nodes cannot be de-activated in later steps, as most IM algorithms consider the progressive models. We limit the discussion about *non-progressive* diffusion models and the corresponding IM algorithms to Section 2.1.5.

Different models apply different mechanisms to capture how a user switches its status from *inactive* to *active*, which is influenced by its neighbors. This section only focuses on four representative models that are commonly used in the IM problem, namely *Independent Cascade (IC)* model, *Linear Threshold (LT)* model, *Triggering (TR)* model, and *Time Aware* model. We also briefly discuss typical non-progressive diffusion models.

2.1.1 The Independent Cascade (IC) Model

Independent Cascade (IC) is a classic and well-studied diffusion model [33]. It considers a user v is activated by each of its incoming neighbors independently by introducing an *influence probability* $p_{u,v}$ to each edge $e = (u, v)$. Based on the influence probabilities and given a seed set S at time step 0, a diffusion instance of the IC model unfolds in discrete steps. Each active user u in step t will activate each of its outgoing neighbor v that is inactive in step $t - 1$ with probability $p_{u,v}$. The activation process can be considered as flipping a coin with head probability $p_{u,v}$: if the result is head, then v is activated; otherwise, v stays inactive. Note that u has *only one chance* to activate its outgoing neighbors. After that, u stays active and stops the activation. The diffusion instance terminates when no more nodes can be activated. The influence spread of seed set S under the IC model is the *expected* number of activated nodes when S is the initial active node set and the above stochastic activation process is applied.

Determining influence probabilities. Some early works rely on heuristic probability assignment. A commonly-used one is weighted cascade (WC) [50]. It assigns $p_{u,v}$ on edge $e = (u, v)$ as $1/d_v^{in}$, where d_v^{in} is the in-degree of v .

Recently, some studies propose to *learn* influence probabilities from data, e.g., propagation actions (e.g. replies, for-

wards, etc.) in the social networks [36], [59], [76], [92]. Saito et al. [92] are the first to formalize the problem of learning edge probabilities from past propagation actions as a likelihood maximization problem. Given a graph $G = (V, E)$ and a set of independent propagation actions, they adopt the *Expectation Maximization (EM)* algorithm to iteratively compute the propagation probabilities for all $e \in E$ to maximize the total likelihood of all actions. Subsequently, Mathioudakis et al. [76] propose the SPINE algorithm to learn the social graph structure and the propagation probability simultaneously where the optimal parameters maximize the log likelihood of generating the propagation actions. Goyal et al. [36] also study the problem of learning edge probabilities and propose more scalable algorithms. Kutzkov et al. [59] further consider this problem in data streams. They propose efficient algorithms to estimate the probabilities with only one pass over all actions. There are some existing works on understanding the model learnability. For example, [79] applies the Probably Approximately Correct (PAC) framework to analyze the diffusion models’ learnability. In [82], an information-theoretical lower bound for the number of cascades needed to learn the IC model is established.

2.1.2 The Linear Threshold (LT) Model

Linear Threshold (LT) is also a seminal diffusion model, which is introduced by Granovetter and Schelling [39], [94] in 1978. The basic idea of LT is that a user can switch its status from *inactive* to *active* if a “sufficient” number of its incoming neighbors are active.

Formally, in the LT model, each edge $e = (u, v) \in E$ is associated with a weight $b_{u,v}$. Let $\mathcal{N}_I(v)$ be the set of incoming neighbors of user v , and it satisfies that $\sum_{u \in \mathcal{N}_I(v)} b_{u,v} \leq 1$. Moreover, each user v is also associated with a threshold θ_v . Considering an instance of the diffusion process, the LT model first samples the value of θ_v of each user v uniformly at random from $[0, 1]$. Then, it proceeds in discrete steps. In step 0, it sets the status of users in S as active and others as inactive. Then, it updates the status of each user iteratively: In step t , all users that were active in step $t - 1$ remain active, and any user v that were inactive in step $t - 1$ switches to active if the total weight of its *active neighbors* in $\mathcal{N}_I(v)$ is at least θ_v . The diffusion instance terminates when no more user is to be activated. Given multiple instances of the diffusion processes, the influence spread of seed set S under the LT model, i.e., $\sigma(S)$, is the expected number of activated nodes when S is initially activated.

Most IM algorithms use heuristics to assign the weight $b_{u,v}$ for each edge $e = (u, v) \in E$, e.g., uniformly assigning e with a probability from the set $\{0.1, 0.01, 0.001\}$ at random or using a similar method to the WC model ($1/d_v^{in}$) [37]. To the best of our knowledge, there is no data-driven approach to assign the probabilities for the LT model.

2.1.3 The Triggering (TR) Model

Kempe et al. [50] propose the triggering model (TR) to generalize the aforementioned IC and LT models. Given any user v , the TR model defines a distribution that maps a subset of v ’s neighbors to a probability, which represents the likelihood that the neighbor subset can influence v . For each instance of the diffusion process, the TR model independently chooses a random “triggering set” T_v for

user v according to the aforementioned distribution over subsets of v 's neighbors. Then, it proceeds in discrete steps. The diffusion instance is again initialized by a seed set S . After the initialization step, an inactive node v switches to the *active* state in step t if it has a neighbor in its chosen triggering set T_v that is activated in step $t - 1$. Similar to IC and LT, the influence of S under TR is also the expected number of activated nodes. It has been proved that both IC and LT are special cases of the TR model [50].

There are more general models than TR that extend IC and LT. For example, [51] extends the IC model to a Decreasing Cascade (DC) Model. DC defines the influence probability from node w to node v given a subset S of active neighbors of v as $p_v(w, S)$. To capture the diminishing return phenomenon, DC enforces $p_v(w, S) \geq p_v(w, T)$ for $S \subseteq T$. Kempe et al. [50] propose a General Threshold (GT) model that extends IC and LT. GT defines the threshold function of v to be $f_v(S)$ where S is the active neighbors of v . Whenever $f_v(S)$ exceeds the threshold value θ_v , v becomes active in the diffusion process. Although the GT model exhibits inapproximability for IM in the most general setting, [78] proves the influence function of GT has the same monotone and submodular properties as those of IC, LT and TR, when the threshold function is monotone submodular and the thresholds are chosen uniformly at random. Nevertheless, since the most common models adopted in existing IM algorithmic researches are IC, LT and TR, we limit the discussions on DC and GT onward.

2.1.4 Time-aware Diffusion Models

IC, LT and TR are *time-unaware* models where the diffusion terminates only when no more node could be activated. However, propagation campaigns are often time-critical and require to maximize the influence spread under a time constraint. To meet such demand, *time-aware* models are proposed, and the existing studies can be broadly classified into two categories: 1) the *discrete-time* models where diffusion only happens in discrete steps, and 2) the *continuous-time* models where the process of one user influencing another (i.e., the diffusion) is continuous in time.

The *discrete-time* models [14], [60], [71] extend IC by modeling the diffusion process from one node to another as a discrete random variable over different time steps. Nevertheless, these models are essentially similar to IC and LT, as the diffusion only happens in discrete steps.

In real-world scenarios, the process of one node influencing another is inherently continuous in time. To capture such essence, continuous-time diffusion models are introduced in [90], [111]. The Continuous-Time (CT) IC model [90] considers the likelihood of pairwise propagation between nodes is a continuous distribution of time. Specifically, given the activation time t_u of a node u , the conditional likelihood of u activating its neighbor v at any time $t_v > t_u$ is defined as $p(t_v | t_u; \alpha_{u,v})$ where $\alpha_{u,v}$ is the parameter of a time-aware influence distribution to determine the influence strength from u to v . Given a predefined stopping time $T > 0$, each diffusion instance of CT stops when no more node is activated before T . A typical choice of the time-aware influence distribution is the exponential model, i.e., $p(t_v | t_u; \alpha_{u,v}) = \alpha_{u,v} \cdot e^{-\alpha_{u,v}(t_v - t_u)}$ if $t_v > t_u$ and 0 otherwise. The DynaDiffuse model [111] considers the propaga-

tion rates of nodes decrease exponentially over time. Given a node u activated at time t and an edge $(u \rightarrow v)$ with propagation rate $r(u, v)$, the propagation probability from u to v at time t' ($t' > t$) is $1 - e^{-r(u,v) \cdot (t' - t)}$. Similarly, DynaDiffuse also restricts the diffusion time to a predefined threshold $T > 0$. One can easily show the equivalence between DynaDiffuse and CT. There are also some studies on modeling the temporal dynamics of diffusion process [93], [104]. However, the focus of these works is to understand the temporal influence behaviour from observation data and, to the best of our knowledge, no IM algorithm is proposed based on these models. Since this survey emphasizes the algorithmic aspect of IM and CT is the most widely adopted time-aware model in IM algorithmic research, we focus on comparing different IM algorithms under the CT model.

2.1.5 Non-Progressive Diffusion Models

There are also several diffusion models that are categorized as *non-progressive* models. The major difference between progressive and non-progressive models is that activated nodes can be de-activated in *non-progressive* models. Typical *non-progressive* diffusion models are the SIR/SIS model [52] and the Voter model [21]. Some IM algorithms are also proposed under the Voter model [27], [66] and the SIR/SIS model [91], [110]. In the remaining of this paper, we will not further discuss *non-progressive* models and the IM algorithms under such models as the focus of our survey is to review the algorithmic aspects of IM where the generally recognized models in this area are LT, IC, TR, and CT.

2.2 Problem Hardness of Influence Maximization

Now, we are ready to discuss the computational hardness of IM under the above-defined diffusion models, i.e. IC, LT, TR and CT.

Theorem 1. The influence maximization problem is NP-hard under the IC, LT, TR and CT models.

Due to space limit, we briefly introduce the proof sketch of Theorem 1. To prove the NP-hardness of IM under the IC model, the idea is to reduce from the *set cover* problem to IM, whereas the idea for the proof under the LT model is to reduce from the *vertex cover* problem. It is then straightforward to extend the hardness result of IM to the TR and CT models since IC and LT are special cases of TR and CT. The detailed proofs can be found in [50].

According to Definition 2, a fundamental operation in IM is to evaluate the influence $\sigma(S)$ of a seed set $S \subseteq V$. To fully understand the complexity of IM, existing researches have established the complexity for the influence evaluation under the IC and LT models.

Theorem 2. [17, Theorem 1] Computing the influence $\sigma(S)$ of a seed set S is #P-hard under the LT model.

Theorem 3. [15, Theorem 1] Computing the influence $\sigma(S)$ of a seed set S is #P-hard under the IC model.

Theorem 2 can be proved by reducing from the problem of counting simple paths in a directed graph. Theorem 3 can be proved by reducing from the counting problem of $s - t$ connectness in a directed graph [103]. As mentioned earlier, IC and LT are special cases of TR and CT. Thus, it is easy

Algorithm 1: GREEDY (k, σ)

Input : k : A Number; $\sigma(\cdot)$: Influence Function.
Output: S : Seed Set.

```

1  $S \leftarrow \emptyset$ 
2 for  $i = 1, \dots, k$  do
3    $u^* \leftarrow \arg \max_{u \in V \setminus S} (\sigma(S \cup \{u\}) - \sigma(S))$ 
4    $S \leftarrow S \cup \{u^*\}$ 
5 return  $S$ 

```

to verify that computing the influence is also #P-hard under the TR and CT models.

Given the above theorems, we know that there is no algorithm to obtain its optimal solution in polynomial time unless $P = NP$. Moreover, even the evaluation of influence spread $\sigma(S)$ is also very complex. Thus, existing research efforts have focused on devising efficient approximation algorithms for IM. We will focus on reviewing these algorithms in the following sections.

3 OVERVIEW OF IM ALGORITHMS

Although the IM problem is computationally complex in general, the optimal solution can be approximated if the influence function $\sigma(\cdot)$ satisfies two properties, *monotonicity* and *submodularity*, which are formally defined as follows.

Definition 3. An influence function $\sigma(\cdot)$ is monotone iff $\sigma(S) \leq \sigma(S')$ for any $S \subseteq S' \subseteq V$.

Definition 4. An influence function $\sigma(\cdot)$ is submodular iff $\sigma(S \cup \{v\}) - \sigma(S) \geq \sigma(S' \cup \{v\}) - \sigma(S')$ for any $S \subseteq S' \subseteq V$ and $v \in V \setminus S'$.

Intuitively, the monotonicity means that adding more nodes to a seed set S does not reduce its influence spread, while the submodularity can be understood as diminishing marginal gains of the influence spread. Existing researches have validated the monotonicity and submodularity of the diffusion models for IM.

Theorem 4. The influence functions under the IC, LT, TR and CT models are monotone and submodular.

We refer the proof of Theorem 4 to [50] (for the IC, LT and TR models) and [34] (for the CT model). Theorem 4 reveals a fact that, although the optimal solution of IM is intractable unless $P = NP$, one can leverage the *monotonicity* and *submodularity* to provide approximate solutions for IM efficiently with theoretical soundness.

3.1 The Greedy Framework

Most of the existing IM algorithms apply a simple *greedy framework*, which is illustrated in Algorithm 1. The algorithm is initialized with an empty seed set S , and it iteratively selects a node u into S if u provides the maximum marginal gain to the influence function $\sigma(\cdot)$ wrt. S (Line 3). The algorithm terminates when there are k distinct nodes in S .

The theoretical guarantee of the greedy framework depends on whether $\sigma(\cdot)$ is a non-negative monotone submodular function. Fortunately, such condition holds under

the classical models as stated in Theorem 4. Given a non-negative monotone submodular function, Theorem 5 states the approximation ratio of the greedy framework.

Theorem 5. [81, Theorem 2.2] Let $S^* = \arg \max_{|S| \leq k} \sigma(S)$ be the set maximizing $\sigma(S)$ among all sets with size at most k . If the influence function $\sigma(\cdot)$ is monotone and submodular and $\sigma(\emptyset) = 0$, then for the set \hat{S} returned by GREEDY(k, σ) of Algorithm 1, we have: $\sigma(\hat{S}) \geq (1 - (1 - \frac{1}{k})^k) \sigma(S^*)$.

In the literature, the approximation ratio is often simplified as $1 - 1/e$ since $1 - 1/e < 1 - (1 - \frac{1}{k})^k$ for $k > 0$ and $\lim_{k \rightarrow \infty} 1 - (1 - \frac{1}{k})^k = 1 - 1/e$, where e is the base of natural logarithm. Moreover, there is an additional term ε in the approximation ratio for IM algorithms, i.e., $(1 - 1/e - \varepsilon)$. This is because evaluating $\sigma(\cdot)$ is #P-hard, as mentioned in Section 2.2, and thus is often approximated by sampling methods. The term ε accounts for the sampling error.

Extension. The greedy framework shown in Algorithm 1 can be naturally extended to a scenario where the costs of selecting nodes are non-uniform. More specifically, each user u has a cost of $c(u)$ to be selected, and the objective is to select a seed set that maximizes the influence spread while keeping the total cost bounded by a budget B , i.e., $S^* = \arg \max_{\sum_{u \in S} c(u) \leq B} \sigma(S)$. In this non-uniform scenario, the greedy framework can be adapted by simply changing the node selection criterion to be cost-effective (Line 3 of Algorithm 1), i.e., $u^* \leftarrow \arg \max_{u \in V} \frac{\sigma(S \cup \{u\}) - \sigma(S)}{c(u)}$, and adding more users into the seed set as long as the total cost does not exceed the given budget B . Although this approach could provide arbitrary bad solutions, comparing it with the solution returned by running naïve greedy selection until the cost has been exhausted (treating the cost for each node is 1) yields a $\frac{1}{2}(1 - \frac{1}{e}) - \varepsilon$ approximation ratio [54], [58], [62].

3.2 Taxonomy of Existing IM Algorithms

Although the aforementioned greedy framework has a good approximation ratio of $(1 - \frac{1}{e} - \varepsilon)$, IM is still very challenging to solve, because evaluating $\sigma(\cdot)$ is a #P-hard problem even under simple models as stated in Theorems 2 and 3. The theoretical hardness has triggered extensive researches on designing efficient IM algorithms in recent years. We classify existing IM algorithms into three categories, the *simulation based* approach, the *proxy based* approach and the *sketch based* approach, as shown in Figure 2. This taxonomy is based on how an algorithm overcomes the #P-hardness of evaluating the influence function $\sigma(\cdot)$. This section will introduce high-level ideas of these three categories of IM algorithms as well as their pros and cons.

Simulation-based approach. The key idea of this approach is to perform *Monte-Carlo (MC) simulation* for evaluating influence spread $\sigma(S)$ of any seed set S . An example of MC simulation under IC model is illustrated as follows. It starts from seed set S and traverses G by removing each edge $e = (u, v)$ with a probability $1 - p_{u,v}$ until no user can be reached, resulting in a sample instance. In such a way, we can generate multiple sample instances, and the influence spread $\sigma(S)$ can be estimated from the sample instances. Based on the MC simulation, the approach preserves the native way of evaluating the influence, and focuses on

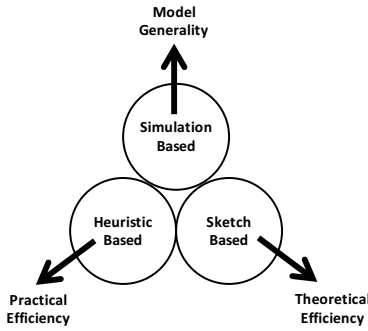


Fig. 2. Taxonomy of IM techniques

using meta-heuristic to speed up the combinatorial optimization of selecting the seed set. Existing algorithms of the simulation-based approach are reviewed in Section 4.

Pros: The simulation based approach has the advantage of *model generality*. In other words, it can easily incorporate any diffusion models mentioned in Section 2.1 with ease by plugging in the model-specific MC simulation module to evaluate the influence. Moreover, the approach has a good theoretical property that it usually returns a solution with a constant bounded ratio of approximation if the underlying influence function is monotone and submodular¹.

Cons: A major problem of the simulation based approach is *computational efficiency*. To overcome the #P-hardness of evaluating influence function $\sigma(\cdot)$, this approach has to generate many sample instances to obtain an estimation of $\sigma(\cdot)$ with small error, which incurs significant computational overheads.

Proxy-based approach. The idea of this approach is to devise proxy models to approximate influence function $\sigma(S)$ for overcoming the #P-hardness. Theoretically, evaluating $\sigma(S)$ is complex as S can potentially influence other users via a large number of paths in the graph. Nonetheless, this approach believes that the complex influence model can be effectively reduced to proxy models, e.g., PageRank [89] or the shortest path [56] in practice. Existing algorithms of the proxy-based approach are reviewed in Section 5.

Pros: The proxy-based approach is *practical efficient*. For example, only considering the shortest path [56], the evaluation of $\sigma(S)$ is polynomial instead of #P-hard. Many algorithms in this approach have shown empirical efficiency superiority of the proxy-based method (see Section 5).

Cons: Although the proxy-based approach usually improves practical efficiency, it lacks theoretical guarantees. It has been shown that under certain circumstances, e.g., an Erdős-Rényi random graph $G(V, E)$ with a sharp threshold of $1/n$, the proxy-based approach is unstable [43] (the optimal seed set and the corresponding influence could drastically change with a minimum change in the underlying graph). Since the proxy-based solutions are often insensitive to the unstable scenarios, they could be arbitrarily bad.

Sketch-based approach. The designing goal of sketch-based approach is to devise *theoretically efficient* solutions (instead of being only *practical efficient*) that also preserves a constant approximation ratio, and thus overcome the drawbacks of the above two categories of approaches. For example, the

expected time complexity to get a solution in this category [100] is near linear to the size of the input graph with a constant approximation ratio. The idea of this approach is to first construct theoretically grounded *sketches* under the diffusion model. Then, the approach speeds up the evaluation based on the constructed sketches to evaluate the influence function. Existing algorithms of the sketch-based approach are reviewed in Section 6.

Pros: The most significant advantage of the sketch-based approach is its theoretical results, i.e., it is the most theoretically efficient algorithm with rigorously bounded solutions and proven low time complexities.

Cons: The sketches constructed must be aligned with the underlying diffusion model. Thus, the theoretical results of the approach are not general to a wider range of diffusion models compared with the simulation based approach. In addition, the practical efficiency of the sketch based approach could be worse than that of the proxy-based approach since it needs to ensure the approximation ratio for the worst case scenario.

In summary, we classify three major categories of existing IM algorithms based on how the approaches improve the IM algorithmic design from three objectives: *model generality* (simulation based), *practical efficiency* (proxy based) and *theoretical efficiency* (sketch based). A theoretical analysis of existing IM algorithms is summarized in Table 1, where details of the algorithms can be found in Sections 4, 5 and 6. In the first column of Table 1, we categorize all compared algorithms according our taxonomy. From column 3 to column 6, we indicate whether the compared algorithms support different diffusion models (“✓” for support, “X” for not support, and “?” for maybe support, but not clearly stated). In columns 7 and 8, we give the expected and/or the worst-case complexity of the algorithms respectively. In column 9, we state the approximation ratios of the algorithms for IM (“N.A.” for no approximation ratio is guaranteed; for proxy-based algorithms, the given approximation ratios are for their proxy models). In the following sections, we will review each approach category in detail.

4 SIMULATION-BASED APPROACH

This section presents the IM algorithms which fall into the category of the simulation-based approach. As mentioned in Section 3, this approach utilizes native *Monte Carlo (MC) simulations* for estimating influence function $\sigma(\cdot)$ so as to preserve model generality, and integrates the MC simulations into the greedy framework in Algorithm 1. This section first introduces a basic framework of the simulation-based approach, and then reviews the existing optimization techniques for improving the performance.

Basic Framework. The seminal work on the simulation-based approach is proposed by Kempe et al. in [50] and it extends the greedy framework. It also iteratively selects a node u into S if u provides the maximum marginal gain. Nevertheless, the key difference is that it employs MC simulations for estimating influence spread $\sigma(S \cup \{u\})$ for each user set $S \cup \{u\}$ in G . In particular, an instance of MC simulation, it always starts from $S \cup \{u\}$ in G , simulates the activation process wrt. the corresponding diffusion model, and outputs the number of activated users denoted by

1. The only exception is the Simulated Annealing meta-heuristic based algorithm in [47].

TABLE 1
A comparison of influence maximization algorithms under classical diffusion models

Category	Method	IC	LT	TR	CT	Expected Complexity	Worst-Case Complexity	Approximation	Notes
Simulation	GREEDY [50]	✓	✓	✓	✓	$O(krm\sigma)$	$O(krm)$	$1 - 1/e - \varepsilon(r)$	$O(\sigma)$ is the expected complexity of a simulation, where σ is the influence spread of the seed set.
	CELF [62]	✓	✓	✓	✓	$O(krm\sigma)$	$O(krm)$	$1 - 1/e - \varepsilon(r)$	n' is the number of simulations at each iteration.
	CELF++ [38]	✓	✓	✓	✓	$O(krm\sigma)$	$O(krm)$	$1 - 1/e - \varepsilon(r)$	
	UBLF [114]	✓	✓	?	?	$O(krm\sigma)$	$O(krm)$	$1 - 1/e - \varepsilon(r)$	
	CGA [109]	✓	×	×	×		$O(m + rm_p(n(Z - M) + k(M + n_p)))$	$1 - e^{-\frac{1}{1+\delta\rho}}$	n_p and m_p are the number of nodes and edges in the largest community; Z and M are the number of communities before and after the combination phase. ρ is the pre-defined threshold determining the tightness of the extracted community and δ is a parameter representing the accuracy loss of influence estimation caused by partitioning the graph.
Proxy	SA [47]	✓	×	×	×		$O(r^2m)$	N.A.	T is the number of iterations in SA
	DEGDIS [16]	✓	✓	✓	✓		$O(k \log n + m)$	N.A.	r is the same value among the simulation based approaches; t is the size of the sampled set on G .
	SPIN [80]	×	✓	×	×		$O(tr(n + m) + n \log n + kn)$	N.A.	
	SPIM [57]	✓	×	×	×		$O(knm)$	$1 - 1/e^a$	
	PMIA [15]	✓	×	×	×		$O(n t_{i\theta} + k n_{o\theta} n_{i\theta} (n_{i\theta} + \log n))$	$1 - 1/e^a$	$t_{i\theta}$ is the time complexity for constructing the MIA for one node; $n_{i\theta}$ and $n_{o\theta}$ are the maximum size of one single MIA and MIOA.
	IPA [55]	✓	×	×	×		$O(\frac{n_{i\theta} n_{v_{i\theta}}}{c} + k^2 (\frac{O_{v_{i\theta}} n_{v_{i\theta}}}{c} + (c - 1)))$	N.A.	O_v is the maximum number of nodes having influence paths from one node; $n_{v_{i\theta}}$ is the maximum number of influence paths between any two nodes; c is the number of parallelized processes; We consider only $O(k)$ nodes are kept in the CELF queue.
	LDAG [17]	×	✓	×	×		$O(n t_{i\theta} + k n_{o\theta} m_{\theta} (m_{\theta} + \log n))$	N.A.	$t_{i\theta}$ is the time complexity for constructing the LDAG for one node; $n_{i\theta}$ and m_{θ} are the maximum number of nodes and edges in one LDAG.
	SIMPATH [37]	×	✓	×	×		$O(k \ln P_{\theta})$	N.A.	P_{θ} is the maximum number of simple paths starting from one node. l is the look head value.
	IRIE [48]	✓	×	×	×		$O(k(n_{o\theta} k + m))$	N.A.	$O(m)$ for sparse matrix multiplication; PMIA is adopted to get $AP_S(u)$, $n_{o\theta}$ is the maximum size of one single MIOA.
	IMRANK [19]	✓	×	×	×		$O(n T d_{max} \log d_{max})$	N.A.	T is the number of iterations before convergence; d_{max} is the largest number of paths end in a node with length no more than l (usually $l = 1$).
Sketch	GROUPPR [73]	✓	×	×	×		$O(kmn)(\text{Linear}) O(m + k^2 n)(\text{Bound})$	N.A.	$O(m)$ for the initial influence estimation; For the marginal influence estimation, the Linear approach takes $O(m)$ time per node and the Bound approach takes $O(k)$ time per node.
	EASYIM [30]	✓	✓	×	×		$O(k^2 D(m + n))$	N.A.	D is the diameter of G .
	NEWGREC [16]	✓	×	×	×		$O(krm)$	$1 - 1/e - \varepsilon(r)$	
	STATICGREEDY [20]	✓	×	×	×		$O(\frac{km n^2 \log(n)}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	PRUNEDMC [85]	✓	×	×	×		$O(\frac{km n^2 \log(n)}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	SKIM [22]	✓	×	×	×		$O(\frac{km n^2 \log(n)}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	RIS [6]	✓	✓	✓	✓		$O(\frac{k(m+n) \log^2 n}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	TIM/TIM + [100]	✓	✓	✓	✓		$O(\frac{k(m+n) \log n}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	IMM [99]	✓	✓	✓	✓		$O(\frac{k(m+n) \log n}{\varepsilon^2})$	$1 - 1/e - \varepsilon$	
	SSA [84]	✓	✓	✓	✓		N.A. ^b	$1 - 1/e - \varepsilon$	
	BKRIS [107]	✓	✓	×	×		$O(\frac{(m+n) R (\log n + \log(\frac{n}{k}))}{\varepsilon^2})$	$1 - 1/e - \varepsilon - \varepsilon'$	$ R $ is the expected size of an RR set; ε' is the error bound of bottom-K minHash.

a. The approximation ratio is wrt. the reduced diffusion model.

b. Unknown due to the result in [46]

$I(S \cup \{u\})$. For each $S \cup \{u\}$, the algorithm runs r rounds of MC simulations and takes the average $I(S \cup \{u\})$ as an estimation of influence spread $\sigma(S \cup \{u\})$.

In [50] and many of the follow-up works, the number of rounds r for MC simulations is not theoretically analyzed. These works state that the error will be small enough when r is set to a large number empirically, e.g., $r = 10,000$. Chen et al. [11] analyze the relation between r and the relative error ε , and thus present a theoretical result for the SIMUGREEDY algorithm on the IC and LT models, i.e.,

Theorem 6. [11, Theorem 3.7] With probability $1 - 1/n$, the naive GREEDY achieves a $(1 - 1/e - \varepsilon)$ approximation ratio if the number of MC simulations r is set to $\Theta(\varepsilon^{-2}k^2n \log(n^2k))$ on both the IC and LT models. Thus, the algorithm runs in time $O(\varepsilon^{-2}k^3n^2m \log n)$.

With Theorem 6, the complexities on the TR and CT models can also be easily inferred. The complexity of the naive GREEDY can actually be tightened according to the analysis of Lemma 10 in [100], i.e., $O(\varepsilon^{-2}k^3n^2m \log n / OPT)$ where OPT is the optimal influence spread.

The complexity result shows that SIMUGREEDY is prohibitively expensive against large graphs. This triggers a number of research efforts to optimize the algorithm, which fall into two categories, i.e., reducing the number of MC simulations, and reducing the complexity of MC [38], [47], [62], [109], [114]. Now, we introduce these two categories of optimization techniques as follows.

Reducing number of MC simulations. Some methods have been proposed to estimate an *upper bound* of influence spread $S \cup \{u\}$ in order to prune the ones with insignificant influences. CELF [62] exploits the submodularity of influence functions to estimate upper bounds. The intuition behind CELF is the power law principle: most nodes in a social network have very small influences and thus can be easily pruned at subsequent iterations. More formally, let S_i denote the selected seed set after the i -th iteration, and $\Delta(u|S_i) = \sigma(S_i \cup \{u\}) - \sigma(S_i)$ the *marginal influence* of u wrt. S_i . According to the submodularity of influence function, i.e., $\sigma(S_j \cup \{u\}) - \sigma(S_j) \leq \sigma(S_i \cup \{u\}) - \sigma(S_i)$ for any $S_i \subseteq S_j$, we know that $\Delta(u|S_i)$ is an upper bound for any $\Delta(u|S_j)$ s.t. $S_i \subseteq S_j$. Based on this, CELF first computes $\Delta(u|\emptyset)$ for each user $u \in V$ and selects S_1 . Then, $\Delta(u|\emptyset)$ can be utilized as an upper bound as follows. At each iteration $j = 2, \dots, k$, CELF visits users in $V \setminus S_{j-1}$ in a descending order of their upper bounds of $\Delta(\cdot|S_j)$, and computes $\Delta(u|S_{j-1})$ using MC simulations. Instead of visiting all users, CELF triggers an *early termination* whenever the maximum upper bound of unvisited users is already smaller than the maximum $\Delta(u|S_{j-1})$ of visited users. Then, CELF updates the upper bound of each visited user u as $\Delta(u|S_{j-1})$ and proceeds to the next iteration $j + 1$. CELF does not improve the worst-case time complexity but the early termination heuristic enables an up to 700 times improvement in practical performance compared with the SIMUGREEDY algorithm [62].

CELF++ [38] improves over CELF by further avoiding unnecessary MC simulations. CELF++ computes both $\Delta(u|S_j)$ and $\Delta(u|S_j \cup \{v_j^u\})$ for each user u , where v_j^u is the user with the maximum marginal influence among all the users visited before u . In this way, CELF++ avoids

evaluating $\Delta(\cdot|S_{j+1})$ if $S_{j+1} = S_j \cup \{v_j^u\}$ at the $j + 1$ iteration. However, it has been pointed out in [2] and [75] that CELF++ does not demonstrate significant speedups over CELF empirically.

Note that both CELF and CELF++ have to compute $\Delta(u|\emptyset)$ (i.e., $\sigma(\{u\})$) for every user $u \in V$ at the first iteration. This is a rather expensive process as $O(\varepsilon^{-2}n^2m \log n)$ time is required as indicated in Theorem 6. UBLF [114] proposes a method to quickly obtain an upper bound of $\sigma(\{u\})$ for all $u \in V$ using matrix analysis. Let σ' denote a vector where each element is $\sigma(u)$ for $u \in V$, UBLF derives an upper bound estimation technique: $\sigma' \leq \sum_{i=1}^n PP^i \cdot \mathbf{1}$ where PP is the propagation probability matrix associated with the graph. As PP is a sparse matrix, PP^i quickly converges and becomes insignificant for larger i . Thus, a quick upper bound estimation for σ' is obtained by a few sparse matrix multiplications to avoid the costly initial iteration of CELF/CELF++. Empirically, UBLF reduces more than 95% of the MC simulations in CELF and achieves a speedup of 2x–10x [114]. Although UBLF only verifies the correctness of the upper bound estimation techniques for the IC and LT models, we conjecture that, with a careful design of the matrix PP , the approaches may be extended to more general models.

Reducing MC complexity. The other way to optimize SIMUGREEDY is to reduce the complexity of individual MC simulation. Wang et al. [109] propose an orthogonal approach called Community based Greedy algorithm (CGA) using divide-and-conquer to reduce the complexity for MC simulations. The basic idea of CGA is to partition the graph into communities. Then, it utilizes the influence of each node within its community to decide which nodes are selected as seeds. The advantage of CGA is that it only runs MC simulations on local subgraphs. Moreover, it produces a $(1 - e^{-\frac{1}{1+\delta\rho}})$ approximate solution for IM [109] where ρ is the pre-defined threshold determining the tightness of the extracted community and δ is a parameter representing the accuracy loss of influence estimation caused by partitioning the graph.

Summary. To conclude the simulation based approach, we would like to point out that, although numerous efforts in this category have been developed to improve the efficiency of the SIMUGREEDY algorithm, significant computational overhead is still required when extracting the seeds on graphs with billions of edges, as reported by existing experimental results (e.g., the experiments of [2], [100]). The major reason is that the simulation-based approach treats the MC simulation as a black box, which is a double-edged sword: it retains the model generality but prevents further performance improvements through analyzing and optimizing the influence evaluation process by utilizing the properties of diffusion models directly. Thus, more recent IM techniques have started to explore the proxy-based and the sketch-based approaches.

Discussion. Among all simulation based algorithms, the algorithm proposed by Jiang et al. in [47] is an exception as no approximation ratio is ensured. This is due to the fact that a Simulated Annealing (SA) meta-heuristic is used to explore the search space of selecting k seeds out from the entire node set V . Although there is no theoretical guaran-

tee, SA can escape from the local optimum compared to the GREEDY algorithm. The experiments show that SA could produce a seed set with slightly better quality with less running time than the other simulation based approaches [47].

5 PROXY-BASED ALGORITHMS

This section reviews existing IM techniques which fall into the category of the proxy-based approach. The key idea is to estimate the influence spread of the seed set using proxy models instead of running heavy MC simulations and thus to make IM algorithms more scalable on larger graphs. Most proxy approaches are tailored for a specific diffusion model (i.e., the IC/LT model) and the influence estimation process is greatly accelerated by taking advantage of the properties of the corresponding models. Although the proxy-based approaches typically do not provide any theoretical guarantees, they offer substantial performance improvements compared to the simulation based approaches. Empirically, the solutions returned by the proxy based approaches have competitive quality with those provided by simulation-based approaches in most cases.

Based on properties of the proxy models, we review the proxy based algorithms in two branches, i.e., (1) the *influence ranking proxy*, and (2) the *diffusion model reduction proxy*, which will be described in detail as follows.

5.1 Influence Ranking Proxy

The idea of the influence ranking proxy is quite intuitive. It ranks all users in graph G according to a *metric* approximating their influences, and then simply generates the seed set from the ranking directly. Thus, the essential challenge here is to derive a good ranking metric.

Simple Ranking Proxy. There are some simple ranking proxies that can be easily derived from graph G , such as using degree, PageRank [89], and Distance Centrality [29] to select the seeds. However, they may fail to provide fair solutions for IM due to two drawbacks. First, although, to some extent, they are all related to the social influences, the actual influence spread of a seed set under diffusion models substantially diverges from its degree, PageRank score, or distance centrality (see the experiments in [16], [50]). Second, more importantly, none of these ranking proxies accounts for the *influence overlaps* between different seeds because they simply compute the influence spread as a linear combination of the influences of all individual users in the seed set. Thus, the proxies will severely overestimate influence spread $\sigma(S)$ if the influences of users in S have significant overlaps.

Influence-aware ranking proxy. There have been several methods proposed to overcome the drawbacks of the aforementioned simple ranking proxies. They either adopt a simple discount proxy for influence estimation (DEGDIS [16]) or generalize the PageRank proxy (GROUPPR [73] and IRIE [48]).

The idea of DEGDIS is based on the degree proxy. Nevertheless, the difference is that, when user u is selected into seed set S , the influence scores of u 's neighbors are *discounted* by a factor to account for the influence overlaps. Specifically, the influence of u will be subtracted by 1 if u

is a neighbor of v . The drawback of DEGDIS is that it only considers the influences between neighbors and ignores all indirect influence paths. Thus, its solution quality is usually not competitive with other IM algorithms.

Liu et al. [73] focus on extending PageRank from a single node to a set of nodes called Group-PageRank. The Group-PageRank $GPR(S)$ for a set of nodes S is essentially the sum of the PageRank scores of all nodes in S with a "discount" for the mutual influences between the nodes in S . The Group-PageRank $GPR(S)$ is an upper bound estimation of the influence of S under the IC model. Based on Group-PageRank, Liu et al. propose GROUPPR for IM. GROUPPR also follows the greedy framework: it first computes the PageRank score of each node. Then, it iteratively adds the node with the maximum marginal influence wrt. S into S . GROUPPR proposes two methods to estimate the marginal influence of each node: (1) *Linear*(S, v) recomputes the Group-PageRank $GPR(S \cup \{v\})$ by power iterations in $O(m)$; (2) *Bound*(S, v) utilizes $GPR(S)$ and $PR(j)$ for $j \in S \cup \{v\}$ to derive $GPR(S \cup \{v\})$ directly in $O(k)$. Both methods are much faster than native MC simulations. Between both methods, *Linear* is slower but more accurate and *Bound* is the opposite. Finally, GROUPPR returns the seed set S after k iterations.

To generalize the PageRank proxy, Jung et al. propose the *Influence Ranking Influence Estimation* algorithm (IRIE) [48] for IM under the IC model. IRIE derives a system of n linear equations with n variables to estimate the influence $r(u)$ of each node u in the graph. The idea behind the linear formulas is intuitive: the influence $r(u)$ of a node u comprises its influence to itself, i.e., 1, and the sum of the influences it propagates to all its neighbors, i.e., $\sum_{v \in N_u^{out}} p_{uv} r(v)$, where N_u^{out} is the set of u 's out-neighbors and p_{uv} is the propagation probability from u to v . Through solving the system of linear equations, the influence $r(u)$ of each node u is assigned. IRIE adds the node with the highest influence into the seed set S and updates the formula for each node u as follows: $r(u) = (1 - AP_S(u))(1 + \alpha \sum_{v \in N_u^{out}} p_{uv} r(v))$, where $\alpha \in (0, 1)$ is the damping factor, and $AP_S(u)$ is the probability that u is activated by S . $AP_S(u)$ can be assigned by existing algorithms such as PMIA [15] and native Monte Carlo simulations [50]. IRIE updates and solves the system of linear equations for k times, and retrieves the seed set S after k iterations.

Other ranking proxies. There are also some other influence ranking proxies. Narayanam and Narahari propose the *Shapley value-based Influential Nodes* algorithm (SPIN) for the LT model in [80]. SPIN models users in G as players in a coalitional game and captures diffusion process as the coalition formation in the game. The Shapley value of each user provides the marginal contributions for the diffusion and is approximated by a simulation-based approach. Then, all the users in G are ranked by their Shapley values in a non-increasing order, and the algorithm iteratively adds the top-ranked one which is not adjacent to any node in S to S until k users are chosen. If all nodes are adjacent to at least one node in S , it just picks the top-ranked one.

Cheng et al. propose the IMRANK [19] algorithm for IM under the IC model. They first give the definition of *self-consistent ranking*: A ranking of nodes $r = \{v_{r_1}, \dots, v_{r_n}\}$

is a self-consistent ranking iff $\Delta_r(v_{r_i}) > \Delta_r(v_{r_j})$ for all $1 \leq i < j \leq n$, where $\Delta_r(v_{r_i}) = \sigma(\{v_{r_1}, \dots, v_{r_i}\}) - \sigma(\{v_{r_1}, \dots, v_{r_{i-1}}\})$. IMRANK is an iterative framework to find a self-consistent ranking from any initial ranking. They devise a Last-to-First Allocating (LFA) strategy to estimate the marginal influence $\Delta_r(v_{r_i})$ of each node wrt. ranking r . Theoretically, it is shown that any initial ranking can finally converge to a self-consistent ranking after iteratively performing the computation of marginal influences and re-sorting the nodes in non-increasing order of marginal influence. They also show that a good initial ranking, e.g., PageRank scores based ranking, tends to converge to a “better” final ranking where top nodes have a larger influence. After finding a self-consistent ranking, it returns top- k nodes in the ranking as the solution for IM.

Summary. The advantage of influence ranking proxies is that they efficiently estimate the influence spread by transforming it to easier problems, e.g., PageRank. However, although the computational overhead is largely reduced, the transformed problems may not be directly related to the IM problem. Therefore, the influence estimation may diverge seriously from the actual influence spread under diffusion models. Moreover, the properties of diffusion models are ignored in influence ranking proxies. To solve this problem, some diffusion model reduction proxies are introduced, which will be presented in the next subsection.

5.2 Diffusion Model Reduction Proxy

Difference from the previous influence ranking proxies, *diffusion model reduction* proxy aims to simplify the diffusion process so as to address the #P-hardness of evaluating the influence function $\sigma(\cdot)$. More specifically, there are two main ideas for diffusion model reduction proxy in general: (1) reducing the *stochastic* diffusion models (i.e., IC and LT) to a *deterministic* model where the influence spread of any seed set can be computed exactly, and (2) restricting the influence range of each user u to a small local subgraph G'_u containing u and ignoring the rest. After devising efficient algorithms for computing the proxy model, such approaches employ the greedy framework in Algorithm 1 to provide a solution for IM. As most of the existing studies in this category focus on IC or LT, we next review the proposed proxy tailored for these two models respectively.

5.2.1 IC Model Reduction Proxy

Recall that IC model assigns an influence probability $p_{u,v}$ to each edge $e = (u, v)$ as mentioned in Section 2.1. Thus, in the IC model, a user u_1 may activate another user u_2 through a large number of paths with different probabilities, which increases the complexity of influence estimation. To address this challenge, some IC model reduction proxies are proposed to only consider the *significant* paths.

The first reduction proxy for IC is the *Shortest-Path Model* (SPM) and *SP1 Model* (SP1M) proposed by Kimura et al. in [56]. The idea is to only consider the shortest path from u to v in the activation process. More formally, let $d(u, v)$ be the distance from u to v , i.e., the number of edges in the shortest path from u to v , and let $d(S, v)$ be the minimum distance from any user in S to v , i.e. $d(S, v) = \min_{u \in S} d(u, v)$. Let us consider S be the seed set.

Then, in SPM, S has only one chance to activate v in step $d(S, v)$. SP1M slightly generalizes SPM by considering S can activate v in steps $d(S, v)$ and $d(S, v) + 1$. In such a way, SP1M largely limits the number of influence paths from a node set S to any node v by pruning all paths with lengths larger than $d(S, v) + 1$, and thus the influence $\sigma(S)$ can be exactly and efficiently computed by the Dijkstra shortest-path algorithm. Since the influence function under SPM/SP1M is still monotone and submodular, the greedy framework can also guarantee a $(1 - 1/e)$ approximate solution under both proxy models. However, since SPM and SP1M only consider the length of influence paths and ignore their influence probabilities, it cannot provide good approximate solutions when the edge probabilities are neither constant nor small. This is because the influence between two nodes is small and sensitive (sensitivity means a change in the influence probability of an edge leads to a large change in the influence) and cannot be tightly approximated by the shortest distance.

The MIA/PMIA [15] model is the most well-known reduced model for IC. The main idea of the *maximum influence arborescence* (MIA) model is to restrict the influence diffusion of node u to a local tree structure rooted at u . Since the influence of a node in the tree can be computed efficiently and exactly, the influence estimation becomes much faster and the MC simulations are avoided. Specifically, the MIA model performs the reductions from two aspects: (1) For any pairs of nodes (u, v) , it considers u can only influence v through the *maximum influence paths* (MIP); (2) Given threshold θ , it further ignores all MIPs with the propagation probabilities less than θ . The propagation probability of a path is the product of influence probabilities of all edges in the path. The *maximum influence paths* $MIP(u, v)$ is the path with the maximum influence probability among all paths from u to v . With two reductions, the Dijkstra shortest-path algorithm can be adapted to construct a *maximum influence in-arborescence* $MIIA(v, \theta)$ containing all MIPs ended with v with propagation probabilities at least θ and a *maximum influence out-arborescence* $MIOA(v, \theta)$ containing all MIPs started from v with propagation probabilities at least θ for each node $v \in V$. By using MIIAs and MIOAs, the incremental influence spread $\Delta(u|S)$ of adding any user u to a seed set S can be computed efficiently. Then, Chen et al. further extend it to the *prefix excluding MIA* (PMIA) model. One issue in the MIA model is that a node u will block the influence of another seed u' in $MIIA(v, \theta)$ if u is on the path from u' to v in the in-arborescence. To get a more accurate influence estimation while keeping the in-arborescence structure, PMIA will update the influenced in-arborescence after adding a node into the seed set so that existing seeds will not block the influence of future seeds. As the influence spread in the MIA/PMIA model is monotone and submodular, the greedy framework can also provide $(1 - 1/e)$ -approximate solutions under these proxy models. However, the main drawback of MIA/PMIA is that they will not be scalable if the graph is dense and the edge propagation probabilities are not small. When the graph is dense, limiting the influences to MIPs will incur large errors in influence estimation. For larger propagation probabilities, if θ is small, the MIIAs and MIOAs will become very large and the influence estimation will be slow. Otherwise, the

influence estimation will become inaccurate.

The *independent path algorithm* [55] (IPA) proposed by Kim et al. reduces IC in a similar way to MIA. IPA also prunes all influence paths with the propagation probabilities less than a given threshold. Nevertheless, IPA does not limit the influence paths to MIPs and thus can achieve a slightly better accuracy than PMIA. Assuming the independence of influence paths, IPA treats each path as an evaluation unit and utilizes the well-known OpenMP programming environment [23]. IPA suffers from the same performance issue as PMIA, both of which are not scalable when the graph is dense and the propagation probabilities between nodes are not small. Thus, to reduce the memory usage, IPA only maintains the influence paths for a small subset (i.e., $3k$) of nodes and discard all remaining nodes. However, although partly solving the performance issue, this optimization will potentially degrade the quality of seeds.

5.2.2 LT Model Reduction Proxy

The LDAG [17] algorithm has a similar basic idea to PMIA but is tailored for the LT model. Since the influence $\sigma(v)$ under the LT model can be computed exactly and efficiently in directed acyclic graphs (DAGs), LDAG restricts the influence graph to be a DAG. Specifically, $LDAG(v, \theta)$ of node v is constructed by the Dijkstra shortest-path algorithm and contains nodes that have influences on v with probabilities of at least θ . Then, based on the constructed LDAGs, the influence spread of any node and the marginal influence of a node w.r.t. the seed set can be computed efficiently. However, since finding the optimal LDAG for a node itself is NP-hard, the LDAG algorithm may introduce additional quality losses as the constructed LDAGs are sub-optimal. In addition, the LDAG algorithm constructs LDAGs for all nodes before the influence estimation. The LDAG construction procedure is both computation and memory intensive when the size of the graph is large.

Goyal et al. [37] propose the SIMPATH algorithm for IM under the LT model. SIMPATH is based on a fundamental result: the influence of a set of nodes under the LT model can be computed by enumerating all simple paths starting from every node in the set. Since it is #P-Hard to enumerate all simple paths, SIMPATH restricts the enumeration to a small neighborhood by pruning paths with probability smaller than a threshold θ . The influence $\sigma(u)$ of a node u is computed by enumerating all possible simple paths with probabilities at least θ and summing them up. Then, the influence spread $\sigma(S)$ is computed by summing up the influence of each node $u \in S$ in the subgraph induced by $V \setminus S \cup \{u\}$. Finally, to further improve the efficiency, the SIMPATH algorithm makes two optimizations for the greedy framework. First, to accelerate the influence estimation of all nodes in the first round, it finds a vertex cover set and computes the influences of all nodes in the cover. The influences of the remaining nodes are derived directly. Second, for all subsequent rounds, it picks the top- l most promising seed candidates at the start of an iteration and computes the marginal gain of those candidates in a batch. Different from LDAG, SIMPATH estimates the influence spread on the original graph without enumerating all simple paths in advance. Therefore, SIMPATH often achieves higher time and space efficiency than LDAG. When the size of the

seed set increases, SIMPATH then needs to enumerate a larger number of simple paths and LDAG could outperform SIMPATH under certain cases [75].

Another proxy algorithm based on enumerating simple paths is EASYIM [30] for both IC and LT models. It estimates the influence of each node by counting influence paths within length l and also accounts for the overlaps between different paths for higher accuracy. EASYIM uses an iterative method assembling IRIE for global influence estimation and achieves better performance.

Summary. Diffusion model reduction proxies are directly derived from the diffusion models (i.e., the IC or LT model), and fully utilize the properties of these models for influence estimation. In most cases, they can achieve competitive quality with simulation based approaches. However, they cannot achieve a balance between accuracy and efficiency when the number of influence paths and the influence range of nodes/sets are large. In addition, diffusion model reduction proxies are often specific for only one model and cannot be generalized to the other models.

6 SKETCH-BASED ALGORITHMS

This section presents existing IM techniques which fall into the category of the sketch-based approach. The main focus of this approach is to improve *theoretical efficiency* of the simulation based methods while preserving the approximation guarantee. Specifically, recall that the bottleneck of the simulation-based approach is rerunning a large number of costly MC simulations for influence spread evaluations of each candidate seed set. To avoid rerunning the MC simulations, the sketch-based approach pre-computes a number of *sketches* based on the specific diffusion model, and then exploits the sketches for evaluating influence spread. Based on how the sketches are generated, we classify the algorithms of this approach into two branches, i.e., forward influence sketch (FI-SKETCH) and reverse reachable sketch (RR-SKETCH). The existing algorithms using these sketches will be reviewed respectively in this section.

6.1 Forward Influence Sketch

The idea of the forward influence sketch (FI-SKETCH) is to construct a sketch by extracting the subgraph induced by an instance of the influence process wrt. the specific diffusion model. Then, it can estimate the influence spread of a seed set S using these subgraphs accurately with theoretical guarantee. Taking the IC model on a graph $G(V, E)$ as an example, a sketch can be constructed by removing each edge $e = (u, v)$ of G with probability $1 - p_{u,v}$ and resulting in a subgraph of G , denoted by G_i . Let us use $I_{G_i}(S)$ to denote the set of users that can be reached by S on G_i . Then, given θ constructed sketches $\{G_1, G_2, \dots, G_\theta\}$ and a seed set S , $\sigma(S)$ is evaluated as the average number of users reached by S on these sketches, i.e., $\sigma(S) = \frac{1}{\theta} \sum_{i=1}^{\theta} I_{G_i}(S)$. The theoretical result shows that the greedy framework with FI-SKETCH as influence estimation can achieve a $(1 - 1/e - \epsilon)$ approximate solution for IM with high probability. Next, we review the algorithms using FI-SKETCH and illustrate their theoretical properties and efficiency in detail.

NEWGREIC proposed in [16] applies FI-SKETCH under the IC model. More specifically, at each iteration of the

greedy framework, NEWGREIC constructs a number of sketches for evaluating $(\sigma(S \cup \{u\}) - \sigma(S))$ for all $u \in V \setminus S$ simultaneously. Note that the asymptotic complexity to construct a sketch under the IC model is the same as running a MC simulation. Thus, NEWGREIC significantly boosts the performance compared to the SIMUGREEDY algorithm described in Section 4 as the sketches constructed are shared by $O(n)$ influence function evaluations.

STATICGREEDY [20] takes another step in the direction of using FI-SKETCH. It constructs θ sketches and uses the sketches to perform all influence evaluations. The following lemma shows the number of sketches required by STATICGREEDY to ensure a $(1 - 1/e - \varepsilon)$ approximate solution with high probability:

Lemma 1. With a probability of $1 - n^{-1}$, STATICGREEDY requires $\theta = (8 + 2\varepsilon) \cdot n \cdot \frac{\log n + \log \binom{n}{k} + \log 2}{\varepsilon^2}$ FI-SKETCH so that a $(1 - 1/e - \varepsilon)$ approximation ratio is achieved.

We can easily deduce Lemma 1 by tweaking the proof in Theorem 1 of [100]. Note that θ could be reduced by using the techniques proposed in [99], but we only present Lemma 1 as the sketch size in the improved version has the same asymptotic complexity. As [20] does not provide a complexity analysis on STATICGREEDY, we show the complexity of STATICGREEDY in the following theorem.

Theorem 7. The time complexity of STATICGREEDY, when θ (Lemma 1) sketches are constructed for solving the IM problem, is $O(\varepsilon^{-2} \cdot k \cdot n^2 \cdot m \cdot \log \binom{n}{k})$

Proof Sketch: STATICGREEDY evaluates $O(kn)$ seed sets using the constructed sketches to extract the seed set. Moreover, to compute $I_{G_i}(S)$ for any sketch G_i and any $S \subseteq V$ where $i = 1, \dots, \theta$, it takes $O(m)$ time in the worst case. Thus, the total complexity for STATICGREEDY is $O(k \cdot n \cdot \theta \cdot m) = O(\varepsilon^{-2} \cdot k \cdot n^2 \cdot m \cdot \log \binom{n}{k})$.

Although STATICGREEDY outperforms the SIMUGREEDY algorithm, its worst-case time complexity is still prohibitively high. A critical reason is that, as shown in the above proof sketch, it takes $O(m)$ to perform any influence evaluation. As a result, various pruning and indexing techniques are proposed to further improve the efficiency, which are presented as follows.

StaticGreedyDU [20] introduces a pruning technique to empirically reduce the running time of STATICGREEDY. The key idea is that, once a seed set S_i is obtained at the end of the i -th iteration of the greedy algorithm, all users reached by S_i on any generated sketches can be pruned, and any subsequent influence evaluations are computed on the pruned sketches, which would improve the performance.

Ohsaka et al. propose PRUNEDMC [85] to further improve the efficiency of StaticGreedyDU by using an index structure on the sketches. For each constructed sketch G_i , PRUNEDMC builds a directed acyclic graph (DAG) and each node in the DAG is a strong connected component on G_i . A hub node is selected for each DAG where the hub node has the maximum degree in the DAG. An index structure is built on each sketch by marking the ancestors and descendants of the hub node on the sketch. To speed up the influence evaluation for any node v , the trick is that if v is the ancestor of a hub for a particular sketch there is no need to traverse the descendants of the hub to know the sets

of users reached by v . Combined this trick with the pruning technique proposed by StaticGreedyDU, the running time to compute the marginal influence for a node v wrt. any candidate seed set can be effectively reduced.

SKIM [22] proposed by Cohen et al. is an interesting approach to speed up the influence estimation on the constructed sketches using bottom- K^2 minHash. The key idea is to perform reverse BFS walks on the sketches and to update the bottom- K minHash values for a number of candidate seed sets at the same time, e.g. evaluating $\sigma(S \cup \{v\})$ for all $v \in V \setminus S$ when S is fixed. SKIM shows significant performance boosts compared with the simulation based approaches and some proxy-based approaches, but its worst-case complexity is still the same with STATICGREEDY as the time to generate the sketch is the bottleneck.

Summary. Although the algorithms mentioned above are based on the IC model, we would like to note that the techniques and theoretical analysis can be easily extended to the LT, TR and CT models. The reason is that these models are all *node-independent* models, i.e., the probability of a user to be activated by its neighbors only depends on its neighbors. In such a way, one can easily extend the FI-SKETCH based approach by constructing the sketches as follows: extracting a subgraph of the entire graph by keeping the incident edges to any node v with a fixed probability distribution indicated by the model. For example, in the LT model, a user v keeps an edge $e = (u, v)$ with probability $b_{u,v}$ in the LT model and keeps no incident edges with probability $(1 - \sum_{(u,v) \in E} b_{u,v})$.

To conclude, the FI-SKETCH approach has significantly improved over the simulation-based approach in terms of efficiency while preserving approximation guarantee. However, the worst-case time complexity is still too expensive to run on graphs with millions of nodes and billions of edges. But the idea of sharing the sketches among the influence evaluations introduced by FI-SKETCH algorithms have opened the gate for developing a more effective type of sketches, i.e., the reverse reachable sketch (RR-SKETCH), which is introduced in the next subsection.

6.2 Reverse Reachable Sketch

Borgs et al. [6] are the first to discover that it is not necessary to estimate the influence using sketches generated by operating on the entire graph. They develop the reverse reachable sketch approach (RR-SKETCH) where the influence of any seed set S is estimated by selecting random nodes and seeing the portion of the randomly selected nodes which can be reached by S . To facilitate our presentation, we first introduce the concepts of the *Reverse Reachable* (RR) set and the *Random RR* set.

Definition 5. Let G' denote a FI-SKETCH constructed on G . The Reverse Reachable (RR) set, i.e., $\mathcal{RR}_{G'}(v)$ for node v contains all the nodes in G' that can reach v . A random RR set, i.e., $\mathcal{RR}(v)$, is generated on an instance of G' sampled from G where v is randomly picked from V .

Intuitively, the random RR set generated from a random node v contains the nodes who can influence v . By building multiple random RR sets on different random nodes, if a node u has a great impact on other nodes, u will have a high

2. Note that this K is different from the size of the seed set k

Algorithm 2: RR-SKETCH (G, k, θ) [100]

Input : $G = (V, E)$: A social graph k : A number;
 θ : Number of RR Sets.
Output: S : Seed Set.

- 1 $\mathcal{R} \leftarrow \emptyset, S \leftarrow \emptyset$
- 2 Generate θ random RR sets and insert them into \mathcal{R}
- 3 **for** $i = 1, \dots, k$ **do**
- 4 Pick node v_i that covers the most RR sets in \mathcal{R}
- 5 Add v_i into S
- 6 Remove from \mathcal{R} all RR sets that are covered by v_i
- 7 **return** S

probability to appear in these random RR sets. Similarly, if a seed set S^* covers the maximal number of the RR sets, S^* is likely to be the optimal seed set. Based on this idea, Algorithm 2 describes the basic framework of the RR-SKETCH approach. RR-SKETCH first generates θ random RR sets (Line 2) and use the standard greedy algorithm on the *maximum coverage* problem [81] to select a seed set S^* of k nodes to cover the maximum number of RR sets (Lines 3-5).

The number of generated random RR sets strikes a balance between efficiency and the solution quality. Borgs et al. [6] propose a threshold-based approach called RIS: they keep generating random RR sets until the total number of edges examined during the generation process reaches a pre-defined threshold τ . They show that when τ is set to $O(\epsilon^{-3} \cdot k \cdot (n + m) \cdot \log^2 n)$, a $(1 - 1/e - \epsilon)$ -approximate solution is returned with probability $1 - 1/n^{-1}$. They later improve their analysis and reduce the complexity to $O(\epsilon^{-2} \cdot k \cdot (n + m) \cdot \log n)$.

To make the RR-SKETCH approach practically efficient, Tang et al. propose TIM [100], which improves over RIS by a better analysis on the number of RR sets required to ensure the same theoretical bound. In particular, it requires $O(\epsilon^{-2} \cdot n \cdot (\log n + \log \binom{n}{k}) / OPT)$ RR sets where OPT is the influence of the optimal seed set. As OPT is an unknown value, [100] proposes a series of bootstrap estimation techniques on OPT and the expected time complexity of TIM is $O(\epsilon^{-2} \cdot (n + m) \cdot \log n)$. By improving the parameter estimation procedure, they also propose the TIM+ algorithm in [100]. TIM+ has the same worst-case complexity as TIM but shows better empirical performance. Tang et al. further propose IMM [99] to improve over TIM/TIM+. IMM uses a martingale analysis and a better bootstrap estimation technique on OPT , which is more efficient than TIM/TIM+.

It is worth noting that the RR-SKETCH approaches, i.e., RIS, TIM, TIM+, and IMM, may have a large memory consumption. There are two reasons for this issue: (1) a large number of samples have to be generated to ensure the theoretical bounds when ϵ is small; (2) all RR sets have to be maintained in the memory to run the greedy algorithm for the seed set selection.

To reduce the memory consumption, Wang et al. [107] propose a lazy sampling technique (BKRIS). First, it utilizes a heuristic method to estimate a lower bound on OPT . Then, a sufficiently large sample size θ is derived from the lower bound on OPT . Given θ RR sets required, BKRIS adopts the bottom-K minHash technique (similar to [22]) and obtains the seed set without fully materializing all θ

sketches unless necessary. It shows that the lazy sampling technique speeds up IMM by two orders of magnitude empirically [107]. The theoretical analysis in [107] shows that BKRIS achieves a $(1 - 1/e - \epsilon - \epsilon')$ -approximation when the bk value for bottom-K minHash is $O(k^2 \epsilon'^{-2} \log n^{2+\log_n k})$ where ϵ is the error of estimating the influence function by RR-SKETCH and ϵ' is the error of estimating each node's coverage by bottom-K minHash. In the experiments, bk is set from 4 to 64. This could violate the theoretical guarantee but demonstrate good empirical results.

An orthogonal “stop-and-stare” optimization SSA proposed in [84] also try to improve over IMM. SSA iteratively doubles the number of sketches and extracts the seeds based on the current generated sketches. Whenever the seed set S_i obtained at iteration i has an estimated influence close to the estimation of S_{i-1} obtained at iteration $i - 1$, it stops and returns the seed set. An improved version of SSA called D-SSA is proposed in [84]. It claims that SSA and D-SSA ensure $(1 - 1/e - \epsilon)$ approximation ratio, but as pointed out in [46], there exist gaps in the analysis of SSA and D-SSA. Although SSA can be fixed to retain the approximation ratio, the theoretical guarantee of D-SSA is difficult to be retained without revising the algorithm substantially [46]. Nevertheless, motivated by SSA and D-SSA, there exist opportunities to further improve the efficiency of IMM while still offering approximation guarantees of the solution [46].

It has been shown that the RR-SKETCH based approach works under the IC, LT and TR models with theoretical guarantees [100]. To extend it to the CT model, Tang et al. propose a shortest-path-like reverse sampling technique in [99]. To sample a RR set under the CT model, the Dijkstra algorithm is invoked to traverse the graph with a random sampled node v as the starting point, following the incoming edges of v only. At each time we encounter an edge e , we sample the length from the influence duration distribution indicated by the CT model. The reverse sampling stops whenever the front element of the Dijkstra distance priority queue is larger than the stopping time T specified by the CT model. It can be shown that the Dijkstra equipped reverse sampling technique has the same asymptotic complexity as IMM. Moreover, this technique could also be applied to RIS, TIM, TIM+, and SSA to support the CT model.

Summary. To conclude, the RR-SKETCH based algorithms are faster than the FI-SKETCH based algorithms in general. The main reason is that each FI-SKETCH is constructed by examining the entire graph whereas constructing a RR-SKETCH only visits the nodes which can activate the randomly sampled node. This results in a major difference in the complexities between two categories of approaches.

7 CONTEXT-AWARE INFLUENCE MAXIMIZATION

The *context-aware* IM studies are emerging in recent years. Extending from the classical IM problem, context-aware IM studies further consider contextual features, such as topic, time and location, in order to “customize” the IM solution to their specific applications. This section reviews IM algorithms under a variety of contexts respectively. Our focus is two-fold: first, we analyze how the existing context-aware IM studies exploit the IM techniques introduced in the previous sections, e.g., diffusion models, IM algorithms,

TABLE 2
Summarization of Context-aware Influence Maximization Algorithms

	Context-aware IM	Diffusion Model	IM Technique	Approximation Bound
Topic	Li et. al [69]	IC	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
	Nguyen et. al [83]	IC	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
	Guo et. al [41]	IC	Model Reduction Proxy	-
	Aslay et. al [3]	IC (topic)	Influence Ranking Proxy	-
	Chen et. al [13]	IC (topic)	Model Reduction Proxy	-
	Chen et.al [10]	MIA (topic)	Model Reduction Proxy	$\epsilon(1 - 1/e)$ wrt. MIA
	Li et. al [68]	IC (topic)	Reverse Reachable Sketch	$(1 - \epsilon)/(1 + \epsilon)$
Location	Li et. al [63]	MIA	Model Reduction Proxy	$\epsilon(1 - 1/e)$ wrt. MIA
	Wang et. al [105], [106]	MIA	Model Reduction Proxy	$(1 - 1/e)$ wrt. MIA
	Song et. al [95]	IC	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
	Zhou et. al [115]	IC	Influence Ranking Proxy	-
Time	Chen et. al [14]	IC-M	Model Reduction Proxy	$1 - \frac{1}{e}$ wrt. MIA
	Liu et. al [71], [72], Lee et. al [60]	LAIC/CT-IC	Model Reduction Proxy	$1 - \frac{1}{e}$ wrt. MIA
	Rodriguez et. al [34]	CT	Continuous Time Markov Chain	$1 - 1/e$
	Du et. al [26], Rodriguez et. al [35]	CT	Forward Influence Sketch	$1 - 1/e - \epsilon$
	Xie et. al [111]	DynaDiffuse	Continuous Time Markov Chain	-
	Tang et. al [99]	CT	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
	Ohsaka et. al [87]	TV-IC/TV-LT	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
Dynamic	Chen et. al [18]	IC	Influence Ranking Proxy	-
	Ohsaka et. al [86]	IC	Reverse Reachable Sketch	$1 - 1/e - \epsilon$
Competitive	Budak et. al [8]	IC (competitive)	Simulation	$1 - 1/e - \epsilon$
	He et. al [45]	LT (competitive)	Model Reduction Proxy	-
	Zhu et. al [116]	IC (competitive)	Simulation	$1 - 1/e - \epsilon$
	Lu et. al [74]	IC (comparative)	Reverse Reachable Sketch	$\alpha \cdot (1 - 1/e - \epsilon)^*$
	Ou et. al [88]	LT (comparative)	Influence Ranking Proxy	-

* α denotes the ratio of the lower bound to the upper bound for the estimated influence under Com-IC in [74]

etc. A summarization of this analysis is illustrated in Table 2 where the details will be explained in the following subsections. Second, we introduce how the context-aware features are integrated into the classical IM problem for novel applications.

7.1 Topic-Aware Influence Maximization

Topic-aware influence maximization (TAIM) extends the generic IM problem by taking the *topics* of the item being propagated into consideration. To formalize the intuition, TAIM introduces *topic* to represent both item characteristics and users' interests, and considers the influence $\sigma(S)$ depends on not only the seed set S but also the topics. Then, given *topics* as a query, it aims at finding the optimal seed set that maximizes the topic-aware influence. We classify the existing TAIM studies into two categories. The first category is *IM for topic-relevant targets*, which considers the *nodes* (i.e., users) is topic-aware, and wants to maximize the influence on a subset of users (called *targets*) relevant to the query topics. The second category is *IM for topic-dependent diffusion*, which formalizes that the *edges* (i.e., user-to-user influence strength) are topic-aware, and wants to maximize the influence under a new topic-dependent diffusion model.

IM for Topic Relevant Targets. Some studies on TAIM [41], [69], [83] focus on maximizing the influence over the users who are relevant to the query topics, i.e., the *topic-relevant targets*. Formally, these studies introduce a concept of *benefit* to differentiate the users. Then, they compute the influence $\sigma(S)$ as the expected summation of benefits of the activated users, which is also known as targeted influence. Based on this, they introduce techniques to find the seed set that maximizes the targeted influence under their benefit computation models.

Li et. al [69] propose to compute a user's benefit by considering how a user matches query topics. More specifically, they associate each user with a profile that consists of the users preferences on different topics. An example

profile of a user is $\{\langle \text{music}, 0.7 \rangle, \langle \text{book}, 0.3 \rangle\}$, which depicts the probabilities that the user likes the topics are 0.7 and 0.3 respectively. Then, given a query topic, say *music*, the benefit of the user is 0.7 as only topic *music* is matched with the query. Given this benefit model, Li et. al [69] address targeted IM problem under the traditional IC model. In general, they adopt the RR-SKETCH framework [6], [100], which is classified by our taxonomy (see Section 6.2). However, the original uniform sampling strategy in RR-SKETCH would not work when considering the benefits. Thus, they introduce a weighted sampling technique to find an unbiased estimator for the targeted influence. Moreover, as conducting online sampling cannot meet the real-time processing requirement, they further devise disk-based index structures to push the sampling procedure from online to offline. The idea is to build a sufficient number of RR sets for each topic (e.g., *music* and *book*) offline. Then, given an online query, it selects RR sets from the query topics and merges the RR sets to compute the result. Li et. al also introduce an incremental index structure to further reduce the I/O cost.

Nguyen et. al [83] generalize the problem by considering any pre-defined benefit function over users. Similar to [69], they also adopt the RR-SKETCH framework [6], [100]. Under the framework, they propose an algorithm with a sampling strategy applicable for general benefit functions, and an early termination rule that avoids generating too many samples. Moreover, this work also studies the cost-aware settings where each user is activated using certain costs.

The personalized IM problem introduced in [41] can be regarded as a special case of the targeted IM. The problem aims to find the seed set that maximizes its influence on one given target user, which can be interpreted that only this user is topic-relevant while the others are not. The work in [41] studies this problem under the IC model, and proposes two algorithms. The first is called efficient local greedy algorithm, which can be classified into *simulation-based* algorithm, with some pruning rules tailored for the local structure of the target user. Obviously, this algorithm

cannot satisfy the online query requirement. The second is an online local cascade algorithm, which is a proxy-based approach that only maintains shortest paths from each user to the target one. However, compared with [69], [83], the influence spread cannot be theoretically guaranteed.

IM for Topic-Dependent Diffusion. Inspired by topic-aware influence analysis [4], [97], recent studies [3], [10], [13], [28], [68] focus on IM for a *topic-dependent diffusion* model. The idea of this new model is to consider each edge $e = (u, v)$ between two users u and v is topic dependent. This is motivated by the fact that v may be activated by u in some topics (e.g., sports) while staying inactive in other ones (e.g., politics). The commonly used topic-aware IC model introduces a topic variable z with values $\{1, 2, \dots, Z\}$, and associates any edge $e = (u, v)$ with Z propagation probabilities $\{p_{u,v}^z\}$. It models a query as a probabilistic distribution over topics $\vec{\gamma} = \{\gamma_1, \gamma_2, \dots, \gamma_Z\}$. Given the query, it computes the propagation probability $p_{u,v}$ of edge e as the dot product $p_{u,v} = \sum_{z=1}^Z \gamma_z p_{u,v}^z$, which can then be used in the standard IC model for computing the influence spread.

The essential challenge here is the enormous number of potential queries, each of which corresponds to a topic distribution and results in a different probabilistic graph. A naive solution is to compute $p_{u,v}$ for each edge given the query and then employ the aforementioned IM algorithms over the induced graph. Obviously, this solution would be very expensive, and thus it calls for efficient approaches to support *online* topic-aware IM queries.

Aslay et al. [3] are the first to study this problem. Their idea is based on an observation that two queries that are similar w.r.t. topic distributions will also have similar influence spread. Technically, they devise an index-based approach INFLEX with *pre-computation* and *similarity search* schemes. It first judiciously samples a set of topic distributions $\Gamma = \{\vec{\gamma}_1, \dots, \vec{\gamma}_m\}$, and pre-computes the seed set under each distribution in Γ by any IM algorithm. At query time, given an online query $\vec{\gamma}_q$, it finds a sufficient set of pre-computed distributions similar to $\vec{\gamma}_q$ from Γ and combines the materialized seed sets by using a *rank aggregation* technique. To this end, INFLEX devises maximum-likelihood Dirichlet estimation for sampling Γ , Bregman-ball tree for fast similarity search, and Kendall- τ distance-based schemes for seed set aggregation. Chen et. al [13] adopt a similar framework and develop optimization techniques, which are suitable for some special graphs with properties like topically-separable and sub-additive.

Chen et. al [10] improve the previous works [3], [13] by providing theoretical guarantee on the influence spread. They adopt the MIA/PMIA model [15] and develop algorithms having a bounded approximation ratio under MIA/PMIA. Specifically, they introduce a *best-effort* framework that estimates an upper bound of the influence for each user and then preferentially computes the exact influence for the users with higher upper bounds so as to prune the insignificant users. Moreover, they also devise a topic-sample-based algorithm that pre-computes seed sets for some offline-sampled topic distributions. However, unlike [3], [13], the algorithm [10] uses the samples to better estimate upper and lower bounds for pruning instead of

directly answering the query, and also achieves theoretical guarantees. Experimental result shows that the work [10] achieves superiority on influence spread while having comparable performance on efficiency with [3], [13].

7.2 Time-Aware Influence Maximization

Classical IM algorithms assume that each diffusion instance stops only when no more node is to be influenced. This assumption is practically unreasonable as the diffusion process may take a long time to stop. For example, it may take $O(n)$ steps in discrete time diffusion models and, in the continuous time models, the process could have an arbitrary time length. Time-aware influence maximization (TimeIM) is thus proposed to impose a time constraint on the diffusion process. First, discrete *time-aware* diffusion models [14], [60], [71], [72] treat the discrete diffusion step as the time measure and restrict the maximal step of the diffusion process. Chen et al. [14] propose the IC-M model where, for each influence edge $(u \rightarrow v)$, u contacts v with a meeting probability $m(u, v)$. Note that u has multiple chances to contact v but u has only one chance to activate v with a probability of $p(u, v)$ if the meeting succeeds for the first time. IM under IC-M finds the optimal seed set which activates the most number of nodes in expectation over random processes of at most τ steps. Liu et al. [71], [72] and Lee et al. [60] propose the LAIC and CT-IC independently in parallel and both models are essentially identical. Given a node u activated in step t and an influence edge $(u \rightarrow v)$, LAIC/CT-IC considers that u activates v in step $t + \delta_t$ with a probability of $p(u, v) \cdot p_u^{lat}(\delta_t)$ where $p_u^{lat}(\cdot)$ is a time delay distribution. Similar to [14], IM under LAIC/CT-IC naturally has a time constraint of τ . As IC-M and LAIC/CT-IC are natural extensions of IC, the IM algorithms under these models are MIA-based solutions where the influence process is simplified to a tree-based diffusion process. As we have discussed, MIA-based solutions uses proxy model techniques and do not have any theoretical guarantees under the original model.

The Continuous-Time Independent Cascade Model (CT) is first proposed in [90]. Since the influence function under CT is still monotone and submodular, [34] uses the greedy framework with the lazy forward optimization for IM under CT. As the CT model can be described as a continuous time Markov chain (CTMC) [34], exactly estimating the influence spread under CT is initially solved as an inference problem for graphical models, which is also #P-hard. Then, [26], [35] introduce a FI-SKETCH based method adapting from SKIM [22] to efficiently evaluate the influence function with a provable guarantee. [99] further introduces a RR-SKETCH based IMM algorithm for IM under CT, which is the first near linear time algorithm with a provable guarantee. [111] proposes the DynaDiffuse model where the edge influence probabilities decay over time. It proposes a CELF-optimized greedy method for seed set selection. But because it cannot provide an error bound for using stochastic model checking on the CTMC Parallel Composition to evaluate the influence function, the proposed method does not have theoretical guarantees. [87] proposes the Time-Varying IC (TV-IC) model to generalize existing time-aware models, e.g., CT and IC-M. Given a node u activated at time t_u and an edge $(u \rightarrow v)$, TV-IC considers the conditional likelihood

that the influence reaches v at time t and v is activated at time t depend on $t - t_u$. [87] also proposes the Time-Varying LT (TV-LT) model which is the first continuous time model extending from LT. Due the submodularity of TV-IC and TV-LT, it proposes a RR-SKETCH based method for IM under both models with a provable guarantee.

We note that TimeIM is a “less researched” area among IM studies and there are still many topics to explore. First, time-aware diffusion models extending from LT/TR are still largely unexplored and [87] is the only work on this topic. Another example is to consider the utilities of users vary with the time periods when they are influenced. It is intuitive that users influenced immediately after the initial seed deployment are more valuable for campaigns in social networks [53].

7.3 Location-Aware Influence Maximization

With the prevalence of location-based social networks (e.g., Twitter, Foursquare, etc.), recent practice of location-aware word-of-mouth marketing has triggered the research interest in *local-aware influence maximization* (LAIM). The basic idea of LAIM is to maximize the influence of the *location-relevant* users, instead of any users in the generic IM settings. To solve this problem, different approaches are proposed to combine the generic IM algorithms with *spatial index* schemes [42], [63], [95], [105], [106], [115].

Li et. al [63] are the first to study LAIM, with a focus on *region* queries: given a geographical region \mathcal{R} , it aims to find a k -sized seed set S that maximizes the influence over \mathcal{R} , i.e., activating the maximum number of users in \mathcal{R} . This work adopts the standard IC model and utilizes the MIA/PMIA model [15] for computing influence spread. It devises a best-first search framework that preferentially accesses the users with large upper bounds and prune unpromising ones with insignificant influence over \mathcal{R} . To this end, it focuses on developing bound estimation techniques for effective pruning. It employs a classic spatial index QuadTree to fast locate the users having influence to the “influencees” in \mathcal{R} . It also develops a *hint-based* algorithm that pre-computes seed set for each leaf region in QuadTree, and combines these seed sets as hints of the regions intersecting with \mathcal{R} for effective upper- and lower-bound estimation. It evaluates the techniques on real location-based social networks, and reports real-time efficiency in milliseconds for various region sizes.

Wang et. al [105], [106] employ a pruning-based framework similar to [63], but focus on *distance-aware query*, which maximizes the influence spread weighted by users’ distance to a query location. Wang et. al [105], [106] also adopt the IC model and the MIA/PMIA proxy [15]. For bound estimation, they judiciously select a set of *anchor locations* and maintain the influence spread of each user given every anchor location as a query. At query time, they can utilize the anchor locations for bound estimation using triangular inequality. This anchor based technique can also fuse with the region-based bound estimation in [63]. Besides this, it also studies marginal influence bound estimation and approximate result estimation for further speedup. Song et. al [95] also study the distance-aware query. Different from [105], [106], they adopt the RR-SKETCH approach. This

work generates a pool of *weighted reverse reachable* (WRR) *trees* and develops a sampling-based approximate algorithm by adapting from the RR-SKETCH approach. Theoretically, it returns a $(1 - 1/e - \epsilon)$ -approximate solution. A proxy-based algorithm is further proposed and it focuses on nodes close to the query location to further improve efficiency. There also exists a study [115] focusing on designing distance-aware weighting model.

Guo et. al [42] introduces an IM problem over trajectory databases: the problem finds k trajectories to be attached with a given advertisement and maximizes the expected influence among a large group of audience. They propose a cluster-based algorithm that partitions the trajectory database into clusters and accesses the clusters in an order such that promising trajectories will be found earlier. Nevertheless, this work is essentially different from the IM studies summarized before, as it does not consider the influence propagation and cannot apply any diffusion models.

7.4 Dynamic Influence Maximization

The IM algorithms discussed so far are inherently static: given a social graph $G = (V, E)$, they assume that G and the propagation probability p_e for any $e \in E$ are fixed. However, real-world social networks keep evolving, e.g., new friendship formed, which continuously affects the influence graph. In the remaining of this subsection, we will introduce major research efforts for dynamic IM, which incrementally process the changes of the social graph.

Given a graph G and the evolution of G during time interval $[t, t+h]$, Aggarwal et al. [1] propose efficient heuristics to find a seed set S at time t such that its influence at time $t+h$ is maximized. Zhuang et al. [117] assume that the changes in the graph can only be detected by *periodically* probing a *small number* of nodes. Based on this assumption, they design efficient algorithms for two problems at each time t : (1) constructing a subgraph \hat{G}_t by probing a set of nodes such that the influence diffusion on the underlying graph G_t at time t can be the best observed; (2) finding a seed set S (using DEGDIS [16]) on the observed subgraph \hat{G}_t to maximize the influence of S on the underlying graph G_t . Note that, the models used in [1] and [117] are simple proxies and do not align with existing diffusion models like IC/LT nor their extensions.

Subsequently, there are several researches [18], [31] modeling the dynamics in the social network as a sequence of snapshot graphs G^1, \dots, G^T . The dynamic IM in this context is to continuously extract the seed set for each snapshot under an diffusion model, e.g., IC. Chen et al. [18] propose an upper bound interchange proxy (UBI). UBI adopts UBLF [114] and SP1M [56] for efficient influence estimation. Then, it tracks the seed set for IM against the up-to-date snapshot graph as follows: (1) use an offline algorithm to retrieve the initial seed set wrt. G^1 ; (2) update the estimated influence spread of each node against a new snapshot; (3) interchange a node into the seed set if it brings a gain of γ (usually 1%) to the total influence spread of the seed set. UBI is 1/2-approximate only if the influence estimation is accurate and any possible interchange is performed as long as it brings any gain. Therefore, UBI has no theoretical guarantee in practice because of inaccurate influence estimation and the threshold for interchange.

Ohsaka et al. [86] first propose a fully dynamic scheme for IM under IC in evolving graphs. Instead of considering snapshot graphs at discrete time steps, their method can provide the seed set for IM in real-time against any node/edge updates. It first constructs an index on RR-SKETCH according to the initial graph. Then, two basic operations, i.e., EXPAND and SHRINK, are proposed to add and delete nodes from sketches by re-sampling. When receiving any change in node/edge, it will update the affected sketches by performing either EXPAND or SHRINK. The core idea of sketch maintenance is to guarantee the probability of sampling any node and edge is always uniformly at random. After the sketch maintenance, it will recompute the sample size θ and generate new sketches or delete existing ones if necessary. Finally, the seed set selection is to perform a maximum k -coverage on dynamically maintained sketches, which is the same as *Phase 2* of static RR-SKETCH based methods like TIM. Although the algorithm is specifically designed for the IC model, the idea may also be extended to other models like LT and TR.

There are some other researches accounting for different concepts of “dynamics”. For example, Lei et al. [61] and Tong et al. [102] are concerned with the incompleteness and uncertainty of the diffusion process. Lei et al. [61] consider the propagation probabilities are unknown in advance and can only be acquired after trials. They propose a method to learn these probabilities at the same time as the diffusion process and adopt ExploreExploit strategies for IM in this setting. Tong et al. [102] consider the propagation probabilities are random variables conforming to certain distributions and propose a simple greedy adaptive seeding strategy to find an effective solution with a provable performance guarantee. Wang et al. [108] study the IM problem over a social action stream. They define the *influence* between users in the sliding window model and propose the *Stream Influence Maximization* (SIM) query to continuously track a seed set maximizing the *influence* wrt. the current window.

7.5 Competitive Influence Maximization

In this subsection, we review the existing research efforts on competitive IM, which consider the scenarios where several competitors spread their influences in the same social network simultaneously and their diffusions interfere with each other. The competitive IM aims to find a strategy for the competitors in a social network such that one’s own influence is maximized while his opponents’ influences are minimized. In the remaining of this subsection, we further categorize existing techniques for competitive IM into three types and review them separately.

Known opponent strategies: Bharathi et al. [5] and Carnes et al. [9] are among the first to formulate the competitive IM with known opponent strategies. They consider the following problem: if there are n players in the diffusion game and the n -th player wants to find the optimal seed set given the choices of the seed sets of the first $(n - 1)$ players. The problem is difficult since if a node is influenced by a player, the node cannot be further influenced by other players. Then, they prove that the greedy strategy achieves the same $1 - 1/e$ approximation guarantee. Borodin et al. [7] propose several extensions of the LT model in the competitive setting. However, they show the proposed models are non-submodular

and it is NP-hard to achieve an approximation that is better than a square root of the optimal solution under these models, where the greedy approach cannot work any more. Subsequently, several techniques are proposed for a variant of competitive IM called influence blocking maximization (IBM). Given a diffusion A of “misinformation” with the seed set S_A , the objective of IBM is to initiate a counter diffusion B with the seed set S_B of size k such that the influence of A is minimized. Budak et al. [8] and He et al. [45] propose greedy solutions for this problem in the IC and LT models respectively. Zhu et al. [116] propose a generalized competitive IC model. They consider a node can serve as the seed for multiple diffusions. The greedy framework is adopted to retrieve a $1 - 1/e - \epsilon$ approximate solution for this problem due to the submodularity of the influence function under their model.

Unknown opponent strategies: In real-world propagation campaigns, it is not practical to assume the opponents’ strategies are known beforehand. Therefore, more practical models where each player does not have knowledge about others’ strategies are proposed for competitive IM. In [70], Lin et al. model the competitive IM problem as a multi-round multi-party game. Li et al. [64] consider another model for competitive IM. Given a graph G and a diffusion model, the strategy space $\Phi = \{\phi_1, \dots, \phi_z\}$ consists of all IM algorithms that may be adopted by players. The objective is to find a Nash equilibrium strategy for each player such that his own influence $\sigma(S_i)$ is maximized. We note that standard IM algorithms are used as building blocks to solve the aforementioned problems.

Comparative IM: The diffusion model for comparative IM considers two different kinds of relationships between two diffusions A and B : (1) Competition: If a node adopts the influence of A , it has a lower probability to adopt B , (2) Complementary: If a node adopts the influence of A , it has a higher probability to adopt B . Lu et al. [74] first propose the Comparative Independent Cascade (Com-IC) model extending the IC model to describe the diffusion of multiple influences with comparative or competitive relationships. Then, they propose the SELFINFMAX problem to maximize the own influence of a diffusion and the COMPINFMAX problem to maximize the incremental influence of a diffusion contributing to another diffusion. They show that both problems are NP-hard and propose RR-SKETCH approaches extending TIM [100] to solve the problems. Ou et al. [88] also consider the comparative IM problem independently. They propose the Interactive Linear Threshold (ILT) model extending the LT model for multiple diffusions. They propose a heuristic strategy TOPBOSS for the second-mover to defeat the first-mover with knowing the first-mover’s seeds selection in the comparative environment.

8 RESEARCH CHALLENGES AND DIRECTIONS

Determining the stability of IM algorithms: He and Kempe [43] show that there is a poor stability of IM algorithms when the input influence probabilities are adversarially noisy. This means a slight change in the diffusion model may change the optimal seed set drastically. Although some efforts have been made to design algorithms for finding

the seed set for robust IM [12], [44], [77], they assume the influence graph structure is fixed. However, the graph structure changes constantly in reality. It is a challenging task to find how the graph structure affects the solution of IM and how to identify a robust seed set given a limited number of graph changes.

Breaking the boundary of submodularity: The submodularity of the influence function plays a vital role for designing efficient and theoretical bounded IM solutions. The submodularity of requirement of the influence function is too strict in certain scenarios. For example, the opinion-aware IM [30], [32], [67] adopts non-submodular influence functions. The non-submodularity occurs as any node can switch between positive and negative opinions which are spread across the influence graph. Under such circumstances, the greedy framework is no longer effective. To provide a better solution than some simple heuristics, a possible future direction is to model the influence function with more general functions, e.g., weakly submodular functions. The weakly submodular functions [24] are more general to model real applications compared to the predominate monotone submodular functions in the IM community. In addition, the theoretical properties of weakly submodular functions guarantee that the extended IM problem can be approximated effectively.

Considering the group norm: Existing IM diffusion models focus on the influence occurred between two nodes with an edge connecting them. In real world, people are not only influenced by acquaintances or friends but are also guided by group norms. One example is the conformity behavior where people often conform within a group, typically of similar age, culture, or educational status. [65], [98] propose a pioneer work of the conformity-aware IM problem. However, the conformity-aware diffusion model defined in [65], [98] considers extracting the conformity characteristics from the graph structure only and thus ignores user profiles from which social groups can be inferred. One possible future direction is to incorporate the user profiles into the conformity IM problem. Moreover, there are different types of conformity: compliance, identification and internalization [49]. These conformity types affect the ways in which people are influenced and it remains a challenging problem on how they can be integrated into IM problems.

9 CONCLUSION

In this paper, we conduct an extensive survey on the IM problem from an algorithmic perspective. We propose a fined-grained taxonomy for classifying existing IM techniques based on their algorithmic designs. We also provide a rigorous theoretical comparative study of existing IM algorithms. Furthermore, we survey the context-aware IM problems and analyze how the IM techniques are used to solve the context-aware IM. We also point out future research challenges. Our survey will give researchers new to IM an understanding of the recent development of IM algorithms and a good starting point to work in this field.

Acknowledgement: Ju Fan was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61602488 and No. 61632016, and Tencent Social Ads Rhino-Bird Focused Research Grant.

REFERENCES

- [1] C. C. Aggarwal, S. Lin, and P. S. Yu, "On influential node discovery in dynamic social networks," in *SDM*, 2012, pp. 636–647.
- [2] A. Arora, S. Galhotra, and S. Ranu, "Debunking the myths of influence maximization: An in-depth benchmarking study," in *SIGMOD*, 2017, pp. 651–666.
- [3] Ç. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries," in *EDBT*, 2014, pp. 295–306.
- [4] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *ICDM*, 2012, pp. 81–90.
- [5] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *WINE*, 2007, pp. 306–311.
- [6] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *SODA*, 2014, pp. 946–957, revision available at <https://arxiv.org/abs/1212.0884>.
- [7] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *WINE*, 2010, pp. 539–550.
- [8] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *WWW*, 2011, pp. 665–674.
- [9] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen, "Maximizing influence in a competitive social network: a follower's perspective," in *ICEC*, 2007, pp. 351–360.
- [10] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, and J. Tang, "Online topic-aware influence maximization," *PVLDB*, vol. 8, no. 6, pp. 666–677, 2015.
- [11] W. Chen, L. V. S. Lakshmanan, and C. Castillo, *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers, 2013.
- [12] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *KDD*, 2016, pp. 795–804.
- [13] W. Chen, T. Lin, and C. Yang, "Real-time topic-aware influence maximization using preprocessing," in *CSoNet*, 2015, pp. 1–13.
- [14] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *AAAI*, 2012, pp. 592–598.
- [15] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD*, 2010, pp. 1029–1038.
- [16] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD*, 2009, pp. 199–208.
- [17] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *ICDM*, 2010, pp. 88–97.
- [18] X. Chen, G. Song, X. He, and K. Xie, "On influential nodes tracking in dynamic social networks," in *SDM*, 2015, pp. 613–621.
- [19] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng, "Imrank: Influence maximization via finding self-consistent ranking," in *SIGIR*, 2014, pp. 475–484.
- [20] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng, "Stat-icgreedy: Solving the scalability-accuracy dilemma in influence maximization," in *CIKM*, 2013, pp. 509–518.
- [21] P. Clifford and A. Sudbury, "A model for spatial conflict," *Biometrika*, vol. 60, no. 3, p. 581, 1973.
- [22] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *CIKM*, 2014, pp. 629–638.
- [23] L. Dagum and R. Menon, "Openmp: An industry-standard api for shared-memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, no. 1, pp. 46–55, 1998.
- [24] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *ICML*, 2011, pp. 1057–1064.
- [25] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD*, 2001, pp. 57–66.
- [26] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *NIPS*, 2013, pp. 3147–3155.
- [27] E. Even-Dar and A. Shapira, "A note on maximizing the spread of influence in social networks," in *WINE*, 2007, pp. 281–286.
- [28] J. Fan, J. Qiu, Y. Li, Q. Meng, D. Zhang, G. Li, K.-L. Tan, and X. Du, "Octopus: An online topic-aware influence analysis system for social networks," in *ICDE*, 2018.

- [29] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978–1979.
- [30] S. Galhotra, A. Arora, and S. Roy, "Holistic influence maximization: Combining scalability and efficiency with opinion-aware models," in *SIGMOD*, 2016, pp. 743–758.
- [31] N. T. Gayraud, E. Pitoura, and P. Tsaparas, "Diffusion maximization in evolving social networks," in *COSN*, 2015, pp. 125–135.
- [32] A. Gionis, E. Terzi, and P. Tsaparas, "Opinion maximization in social networks," in *SDM*, 2013, pp. 387–395.
- [33] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.
- [34] M. Gomez-Rodriguez and B. Schölkopf, "Influence maximization in continuous time diffusion networks," in *ICML*, 2012, pp. 313–320.
- [35] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, and B. Schölkopf, "Influence estimation and maximization in continuous-time diffusion networks," *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 9:1–9:33, 2016.
- [36] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "Learning influence probabilities in social networks," in *WSDM*, 2010, pp. 241–250.
- [37] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *ICDM*, 2011, pp. 211–220.
- [38] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *WWW*, 2011, pp. 47–48.
- [39] M. Granovetter, "Threshold models of collective behavior," *Am. J. Sociol.*, vol. 83, no. 6, pp. 1420–1443, 1978.
- [40] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, 2013.
- [41] J. Guo, P. Zhang, C. Zhou, Y. Cao, and L. Guo, "Personalized influence maximization on social networks," in *CIKM*, 2013, pp. 199–208.
- [42] L. Guo, D. Zhang, G. Cong, W. Wu, and K. Tan, "Influence maximization in trajectory databases," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 627–641, 2017.
- [43] X. He and D. Kempe, "Stability of influence maximization," *CoRR*, vol. abs/1501.04579, 2015.
- [44] —, "Robust influence maximization," in *KDD*, 2016, pp. 885–894.
- [45] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *SDM*, 2012, pp. 463–474.
- [46] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan, "Revisiting the stop-and-stare algorithms for influence maximization," *PVLDB*, vol. 10, no. 9, pp. 913–924, 2017.
- [47] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *AAAI*, 2011, pp. 127–132.
- [48] K. Jung, W. Heo, and W. Chen, "IRIE: scalable and robust influence maximization in social networks," in *ICDM*, 2012, pp. 918–923.
- [49] H. C. Kelman, "Compliance, identification, and internalization three processes of attitude change," *Journal of Conflict Resolution*, vol. 2, no. 1, pp. 51–60, 1958.
- [50] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003, pp. 137–146.
- [51] —, "Influential nodes in a diffusion model for social networks," in *ICALP*, 2005, pp. 1127–1138.
- [52] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 115, no. 772, pp. 700–721, 1927.
- [53] A. Khan, "Towards time-discounted influence maximization," in *CIKM*, 2016, pp. 1873–1876.
- [54] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, 1999.
- [55] J. Kim, S. Kim, and H. Yu, "Scalable and parallelizable processing of influence maximization for large-scale social networks?" in *ICDE*, 2013, pp. 266–277.
- [56] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *PKDD*, 2006, pp. 259–271.
- [57] M. Kimura, K. Saito, and R. Nakano, "Extracting influential nodes for information diffusion on a social network," in *AAAI*, 2007, pp. 1371–1376.
- [58] A. Krause and C. Guestrin, "A note on the budgeted maximization of submodular functions," Carnegie Mellon University, Tech. Rep. CMU-CALD-05-103, 2005.
- [59] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, "Strip: Stream learning of influence probabilities," in *KDD*, 2013, pp. 275–283.
- [60] W. Lee, J. Kim, and H. Yu, "Ct-ic: Continuously activated and time-restricted independent cascade model for viral marketing," in *ICDM*, 2012, pp. 960–965.
- [61] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *KDD*, 2015, pp. 645–654.
- [62] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *KDD*, 2007, pp. 420–429.
- [63] G. Li, S. Chen, J. Feng, K. Tan, and W. Li, "Efficient location-aware influence maximization," in *SIGMOD*, 2014, pp. 87–98.
- [64] H. Li, S. S. Bhowmick, J. Cui, Y. Gao, and J. Ma, "Getreal: Towards realistic selection of influence maximization strategies in competitive networks," in *SIGMOD*, 2015, pp. 1525–1537.
- [65] H. Li, S. S. Bhowmick, and A. Sun, "Casino: Towards conformity-aware social influence analysis in online social networks," in *CIKM*, 2011, pp. 1007–1012.
- [66] Y. Li, W. Chen, Y. Wang, and Z. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," in *WSDM*, 2013, pp. 657–666.
- [67] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," in *WSDM*, 2013, pp. 657–666.
- [68] Y. Li, J. Fan, D. Zhang, and K. Tan, "Discovering your selling points: Personalized social influential tags exploration," in *SIGMOD*, 2017, pp. 619–634.
- [69] Y. Li, D. Zhang, and K. Tan, "Real-time targeted influence maximization for online advertisements," *PVLDB*, vol. 8, no. 10, pp. 1070–1081, 2015.
- [70] S.-C. Lin, S.-D. Lin, and M.-S. Chen, "A learning-based framework to handle multi-round multi-party influence maximization on social networks," in *KDD*, 2015, pp. 695–704.
- [71] B. Liu, G. Cong, D. Xu, and Y. Zeng, "Time constrained influence maximization in social networks," in *ICDM*, 2012, pp. 439–448.
- [72] B. Liu, G. Cong, Y. Zeng, D. Xu, and Y. M. Chee, "Influence spreading path and its application to the time constrained social influence maximization problem and beyond," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1904–1917, 2014.
- [73] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, and J. X. Yu, "Influence maximization over large-scale social networks: A bounded linear approach," in *CIKM*, 2014, pp. 171–180.
- [74] W. Lu, W. Chen, and L. V. S. Lakshmanan, "From competition to complementarity: Comparative influence diffusion and maximization," *PVLDB*, vol. 9, no. 2, pp. 60–71, 2015.
- [75] W. Lu, X. Xiao, A. Goyal, K. Huang, and L. V. S. Lakshmanan, "Refutations on 'debunking the myths of influence maximization: An in-depth benchmarking study'," *CoRR*, vol. abs/1705.05144, 2017.
- [76] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen, "Sparsification of influence networks," in *KDD*, 2011, pp. 529–537.
- [77] Y. Mehmood, F. Bonchi, and D. García-Soriano, "Spheres of influence for more effective viral marketing," in *SIGMOD*, 2016, pp. 711–726.
- [78] E. Mossel and S. Roch, "On the submodularity of influence in social networks," in *STOC*, 2007, pp. 128–134.
- [79] H. Narasimhan, D. C. Parkes, and Y. Singer, "Learnability of influence in networks," in *NIPS*, 2015, pp. 3186–3194.
- [80] R. Narayanam and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 130–147, 2011.
- [81] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
- [82] P. Netrapalli and S. Sanghavi, "Learning the graph of epidemic cascades," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 211–222, 2012.
- [83] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *INFOCOM*, 2016, pp. 1–9.

- [84] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *SIGMOD*, 2016, pp. 695–710.
- [85] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-I. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *AAAI*, 2014, pp. 138–144.
- [86] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Dynamic influence analysis in evolving networks," *PVLDB*, vol. 9, no. 12, pp. 1077–1088, 2016.
- [87] N. Ohsaka, Y. Yamaguchi, N. Kakimura, and K. Kawarabayashi, "Maximizing time-decaying influence in social networks," in *ECML-PKDD*, 2016, pp. 132–147.
- [88] H.-C. Ou, C.-K. Chou, and M.-S. Chen, "Influence maximization for complementary goods: Why parties fail to cooperate?" in *CIKM*, 2016, pp. 1713–1722.
- [89] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," The Stanford InfoLab, Tech. Rep. 1999-66, 1999.
- [90] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *ICML*, 2011, pp. 561–568.
- [91] K. Saito, M. Kimura, K. Ohara, and H. Motoda, "Efficient discovery of influential nodes for SIS models in social networks," *Knowl. Inf. Syst.*, vol. 30, no. 3, pp. 613–635, 2012.
- [92] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *KES*, 2008, pp. 67–75.
- [93] K. Scaman, R. Lemonnier, and N. Vayatis, "Anytime influence bounds and the explosive behavior of continuous-time diffusion networks," in *NIPS*, 2015, pp. 2026–2034.
- [94] T. C. Schelling, *Micromotives and macrobehavior*. W. W. Norton & Company, 1978.
- [95] C. Song, W. Hsu, and M. Lee, "Targeted influence maximization in social networks," in *CIKM*, 2016, pp. 1683–1692.
- [96] J. Sun and J. Tang, "A survey of models and algorithms for social influence analysis," in *Social Network Data Analytics*. Springer US, 2011, ch. 7, pp. 177–214.
- [97] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *KDD*, 2009, pp. 807–816.
- [98] J. Tang, S. Wu, and J. Sun, "Confluence: Conformity influence in large social networks," in *KDD*, 2013, pp. 347–355.
- [99] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD*, 2015, pp. 1539–1554.
- [100] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *SIGMOD*, 2014, pp. 75–86.
- [101] V. Tejaswi, P. V. Bindu, and P. S. Thilagam, "Diffusion models and approaches for influence maximization in social networks," in *ICACCI*, 2016, pp. 1345–1351.
- [102] G. Tong, W. Wu, S. Tang, and D. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 112–125, 2017.
- [103] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM J. Comput.*, vol. 8, no. 3, pp. 410–421, 1979.
- [104] S. Wang, X. Hu, P. S. Yu, and Z. Li, "Mmrates: inferring multi-aspect diffusion networks with multi-pattern cascades," in *KDD*, 2014, pp. 1246–1255.
- [105] X. Wang, Y. Zhang, W. Zhang, and X. Lin, "Distance-aware influence maximization in geo-social network," in *ICDE*, 2016, pp. 1–12.
- [106] —, "Efficient distance-aware influence maximization in geo-social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 599–612, 2017.
- [107] X. Wang, Y. Zhang, W. Zhang, X. Lin, and C. Chen, "Bring order into the samples: A novel scalable method for influence maximization," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 243–256, 2017.
- [108] Y. Wang, Q. Fan, Y. Li, and K.-L. Tan, "Real-time influence maximization on dynamic social streams," *PVLDB*, vol. 10, no. 7, pp. 805–816, 2017.
- [109] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *KDD*, 2010, pp. 1039–1048.
- [110] S. Wen, M. S. Haghighi, C. Chen, Y. Xiang, W. Zhou, and W. Jia, "A sword with two edges: Propagation studies on both positive and negative information in online social networks," *IEEE Trans. Computers*, vol. 64, no. 3, pp. 640–653, 2015.
- [111] M. Xie, Q. Yang, Q. Wang, G. Cong, and G. de Melo, "Dynadiffuse: A dynamic diffusion model for continuous time constrained influence maximization," in *AAAI*, 2015, pp. 346–352.
- [112] M. Ye, X. Liu, and W.-C. Lee, "Exploring social influence for recommendation: A generative model approach," in *SIGIR*, 2012, pp. 671–680.
- [113] H. Zhang, S. Mishra, and M. T. Thai, "Recent advances in information diffusion and influence maximization in complex social networks," in *Opportunistic Mobile Social Networks*. CRC Press, 2014, ch. 2, pp. 37–68.
- [114] C. Zhou, P. Zhang, W. Zang, and L. Guo, "On the upper bounds of spread for greedy algorithms in social network influence maximization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2770–2783, 2015.
- [115] T. Zhou, J. Cao, B. Liu, S. Xu, Z. Zhu, and J. Luo, "Location-based influence maximization in social networks," in *CIKM*, 2015, pp. 1211–1220.
- [116] Y. Zhu, D. Li, and Z. Zhang, "Minimum cost seed set for competitive social influence," in *INFOCOM*, 2016, pp. 1–9.
- [117] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun, "Influence maximization in dynamic social networks," in *ICDM*, 2013, pp. 1313–1318.

Yuchen Li is an assistant professor at the School of Information Systems, Singapore Management University (SMU). Before joining SMU, he was a research fellow in the School of Computing, National University of Singapore (NUS). He received the double B.Sc. degrees in applied math and computer science (both degrees with first class honors) and the PhD degree in computer science from NUS, in 2013 and 2016, respectively. His research interests include graph analytics and heterogeneous computing.



Ju Fan received the B.Eng. degree in computer science from Beijing University of Technology, China in 2007 and the PhD degree in computer science from Tsinghua University, China in 2012. He worked as a research fellow in the School of Computing, National University of Singapore (NUS) from 2012 to 2015. He is currently an associate professor at Renmin University of China. His research interests include social influence maximization, crowdsourcing data management, and big data analytics.



Yanhao Wang is currently a PhD student in the School of Computing, National University of Singapore. He received his B.Eng. and M.Eng. degrees in computer science from Shandong University in 2011 and Renmin University of China in 2014 respectively. His research interests include data stream algorithms, social network analytics and spatial-temporal databases.



Kian-Lee Tan is a professor at the School of Computing, National University of Singapore (NUS). He received his PhD degree in computer science in 1994 from NUS. His current research interests include query processing and optimization in multiprocessor and distributed systems, database performance, data analytics, and database security. He was a co-recipient of Singapore's President Science Award in 2011. He is also a 2013 IEEE Technical Achievement Award recipient. He is an associate editor of the ACM Transactions on Database Systems (TODS) and WWW Journal. He has also served in the editorial board of the Very Large Data Base (VLDB) Journal (associate editor: 2007-2009; editor-in-chief: 2009-2015) and the IEEE Transactions on Knowledge and Data Engineering (2009-2013). Kian-Lee is a member of ACM and IEEE (and IEEE CS).

