A Survey: Models, Techniques and Applications of Influence Maximization Problem

Yilin Zheng
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China
Email: 11510506@mail.sustc.edu.cn

***Abstract*—Influence maximization, defined by Kempe et al. [3] is a hot problem in the field of network analysis. This problem is to target k users as seeds in a network G and then maximize the spread of influence in that network. Many models are proposed to mimic certain behaviours of the social networks according to the observations on the network. Based on these models, various techniques are applied and they can give a feasible approximation solution even for large-scale networks. In this survey, the influence maximization problem will be formulated and some related theorems are proven. Models and techniques will be summarized following the popular datasets and applications. The future direction on this problem will also be talked in the last.**

I. Introduction

With the rapid development of the Internet around the world, social networks become more and more popular. Social networks connect numerous people together within a short period of time and have revolutionized the way people communicate with each other. Information spreads on social networks, ideas and knowledge are shared through social networks, and people can influence others by interaction on social networks. These properties have attracted scientists from sociology, economics as well as computer science. Thus, many problems from social network analysis are studied, such as the diffusion models and the social influence. Diffusion models are studied to model the behaviour of social networks which further help solve some problems based on these models like *influence maximization(IM)* problem which is to find the k most influential nodes in a social network that maximize the spread of the influence. One of the goal to study the social influence is also to solve the IM problem.

Influence maximization problem is typically explained in a marketing scenario. For example, a company develops a new product and managers want to market this product as much as possible, then they initially target a few influential people(might be free product on trial for them), then ideally, these targeted users will recommend the product to their friends and many individuals will try the product through such "word-of-mouth" effect. Such cascades of influence are usually triggered on a social network. Then the goal of IM is simply to maximize the spread of the influence on a network. For the example presented before, the company aims to maximize the profits gained from the new product.

The problem was firstly proposed by Domingos and Richardson [1] [2] as an algorithm problem and then formulated into an optimization problem by Kempe, Kleinberg, and Tardos [3]. Following basic IM problem, variant IM problems are proposed and studied to achieve various goals extended from original IM problem.

For these IM problems, social networks are modelled as a graph and influence propagation follows certain rules defined by the model. There are two most basic and widely-studied models: independent cascade(IC) model of Goldenberg et al. [4] and linear threshold(LT) model of Granovetter [8]. But these two models are too basic and contains less information to design efficient algorithms to achieve the goals. So, later more new models are proposed like the data-driven model of Lin et al. [5], the location-based model of Han et al. [6], voter model of Even-Dar and Shapira [7], and Ising model of Liu et al. [9]. There are also some extended models derived from basic diffusion models which provide more information for algorithm design.

The basic IM problem is $\#P$-hard under certain models(e.g. linear threshold). Based on these models, many efficient algorithms are designed to find the approximate optimal solution. Greedy and greedy-based algorithms can provide a solution within $(1-1/e)$ of the optimal influence spread [3]. Heuristic algorithms(e.g. DegreeDiscountIC of Chen et al. [10]) can return a matchable result with a lower computation on large-scale social networks compared to greedy strategy. Besides, learning methods [11] are being considered to solve these problems under learning models and the structure attribute of a social network, community [12], is also taken into consideration while modelling the social network and designing the methods. With the increment of big data, large-scale social networks analysis becomes hot fields, thus, making algorithms applied to large-scale social networks feasible are studied widely(e.g. [13] [14] [15]).

The datasets studied in the IM problem come from large online social network company or some other websites with social functions and involves various information. The basic information is nodes and edges. Nodes denote individuals and edges between nodes usually represent their relationship. Big companies will use the data collected from their servers to study the IM problem for further applications.

IM is a process to select most nodes as seeds in a social

network which motivates the application on viral marketing [15], rumour blocking [16], online advertisements [17], etc. In viral marketing, most influential people will be selected as targeted users to try new products and these users are considered to influence others to use the products which can finally lead to a widespread of marketing. Rumour blocking pays attention to select a subset of protectors which can block the spread of rumour(negative information) so as to make a reliable social medium. The goal of online advertisements is to identify key influencers who can effectively contribute to the dissemination of information.

A. Organization of This Survey

The rest of this survey includes five sections. In Sec. II, the notation, and definition of the basic concepts of IM will be introduced and the problem will be formulated into a formal definition. Sec. III will give an introduction on models defined for this problem. These models include basic models and some interesting extended models. Following Sec. III, Sec IV will talk about some techniques designed to solve this problem under certain models. Also, datasets commonly used in the studies of this problem are introduced in Sec. V. Sec. VI is about some details of the applications of IM techniques. The potential future directions in IM research will be discussed in Sec. VII. The last section, Sec. VIII is a conclusion for this survey.

II. Problem Formalization

In this section, the basic concepts, the input, the output of IM will be introduced and the complexity of computing this problem will also be provided. Moreover, some variant problems extended from basic IM problems are presented.

A. Notation and Definition

The details of notation and definition of the basic concepts in influence maximization are list in Table I.

TABLE I
NOTATION USED IN IM PROBLEM

Notation	Description
$G = (V, E)$	A social network G with node set V and edge set E
n	Number of nodes in the network G
m	Number of edges in the network G
S	Initial seed set
S_t	Set of activated(influenced) nodes at step t
$\|\cdot\|$	The cardinality of set
k	Number of seeds to be selected
R	Number of rounds of simulations or iterations
$\sigma(\cdot)$	The total number of nodes activated by the process

1) The social networks G can be any graph representing a social network.
2) V are considered as individuals of social network G
3) E can be considered as the relationship or the direction of diffusion of influence.

B. Problem Definition

***Definition** 1 (Influence Maximization):* Consider a social network $G = (V, E)$, the process of IM initially selects k nodes as seed set S, then follows a propagation rule(e.g. propagation probability), at round $t(\leq R)$, there will have some new adjacent nodes been activated and these newly activated nodes form an activated node set S_t. After termination of propagation, the number of nodes in all the activated set $S_i(i \in R)$ will be σ.

The goal of IM is to find the minimal set S that maximize the size of activated node set σ, so the objective functions are:

$$\begin{cases} \operatorname{argmin}_S |S| \\ \operatorname{argmax}_S \sigma(A) \end{cases} \tag{1}$$

where

$$A = \sum_{t=1}^{R} S_t \tag{2}$$

It can be proven later that $\sigma(A)$ is submodular.

C. Monotonicity

***Definition** 2 (Monotonicity):* A set function f is monotone if $f(S) \leq f(T)$ such that $S \subset T \subset U$.

D. Submodularity

***Definition** 3 (Submodularity):* A set function f is submodular if it satisfies

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \tag{3}$$

$\forall v$ and $S \subset T$.

E. Input

The input of IM process is usually a graph representing social networks. Most of time, they can be represented in directed(e.g. [23] [24] [25]) or undirected(e.g. [20] [21] [22]) graph(static graph). However, some studies also use dynamic graph(e.g. Xie et al. [18]) or random graph(e.g. Yagan et al. [19]). The representation of a graph can various such as, sets, functions, adjacent matrices, adjacent list, and incidence matrices, etc. The input, in applications, is usually large-scale social network and consume large amount computation even running efficient algorithms.

Challenge: How to lower the computation of IM algorithms applied to large-scale social networks with guarantees on the result and time-consuming.

F. Output

The output is usually the seed set S when given the number of seed k.

Challenge: How to determine the number of seeds k that gives guarantees on maximal influence diffusion.

G. Complexity

It has been proven that IM is #P-hard problem [3]. But the concrete complexity varies under different models. Under the basic IC or LT model, this problem is NP-hard. There is no polynomial-time approximation for this problem. In the work of Lu et al. [26], they present a proof of the NP-hard of this problem under a *deterministic linear threshold* model referred as Theorem 4. Some studies on IM techniques are heuristic-based which is a trade-off between accuracy and speed. In the paper of Chen et al. [10], they design a heuristic algorithm *DegreeDiscount* on this problem. Different models will affect the design of algorithms and the complexity will differ from each other.

Challenge: How to design models that make algorithm design feasible on large-scale network data.

H. Variant Problems

There are some variant problems extended from basic IM problem such as, *competitive influence maximization*(e.g. Lin et al. [27]), *time-constrained influence maximization*(e.g. Liu et al. [28]), *topic-aware influence maximization*(e.g. Chen et al. [29]), etc. Various extended problems are studied under specific models constructed from some basic models with certain information added.

III. Models

Social networks are observed by scientists and they create various models to mimic the behaviour of the diffusion process observed from social networks. To model the process, some concepts from other fields are introduced and result in effective modelling. This section will introduce some models studied in IM problems including

1) *cascade models*,
2) *threshold models*,
3) *other models*.

A. Cascade Models

Cascade models are considered for diffusion process inspired by the work on interacting particle system [30] [31]. Goldenberg et al. [4] firstly studied IM under cascade models.

In cascade models, beginning from seeds, at each step t, a activated node v can have only one attempt to activate its adjacent non-active nodes with a probability p_v in an arbitrary order. If succeeds, the newly activated nodes will become active at next step $t+1$ and do the same operation to activate the rest non-active nodes. Whether the attempt is successful or not, the activated nodes cannot try to activate a same nodes twice. The process will terminate until there are no more new activations.

1) ***Independent Cascade****:* Independent cascade(IC) model [3] is the simplest cascade model with the probability on activated node v to its adjacent non-active node u, $p_u(v)$ being constant, which is independent of the previous diffusion process. It can be proven in the work of Kempe et al. [33] that the order of attempt to activate the nodes makes no influence of the diffusion result. Using *order-independence* can show the proof as following(referred in [33]):

Proof: Let S denote the set of nodes that have already attempted and failed to active u, and the probability for v to successfully active u is denoted by $p_v(v|S)$. Let $v_1, v_2, ..., v_k$, and $v'_1, v'_2, \cdots, v'_k$ be two different permutations of S, and $T_i = \{v_1, v_2, ..., v_{i-1}\}$, $T'_i = \{v'_1, v'_2, ..., v'_{i-1}\}$. Then

$$\prod_{i=1}^{k}(1 - p_u(v_i|S \cup T_i)) = \prod_{i=1}^{k}(1 - p_u(v'_i|S \cup T'_i)) \quad (4)$$

where $S \cap T = \emptyset$ ■

2) ***Weighted Cascade****:* Weighted cascade(WC) model is studied in by Chen et al. [10]. Different from IC model, weighted cascade model has a changeable diffusion probability related to the previous steps. If in round t, the probability of edge $\overline{uv} \in E$ is $p_u(v)$, and adjacent not-yet-activated nodes are activated by v independently. During the t-th round, if a not-yet-activated node v has ℓ neighbours activated, then the probability that v is activated in round $t+1$ is $1-(1-p_u(v))^{\ell}$. In the work of Wei Chen, they use this model with directed graph and apply a greedy algorithm *NewGreedyWC* to this model.

3) ***Decreasing Cascade****:* Decreasing cascade(DC) is proposed in the work of Kempe et al. [33]. In this model, a not-yet-activated node v will suffer "marketing-saturated" if more nodes have attempted to activate v. Then the independent cascade model turns to be a special case of decreasing cascade which the not-yet-activated nodes will not suffer a "marketing-saturated".

4) ***IC with Negative Opinion****:* If considers negative propagations in the diffusion process, that is a totally different situation. In the work of Chen et al. [34], they proposed an IC model with *negative opinion*(IC-N) considered in the diffusion process. In this model, a new parameter q, called *quality factor*, is introduced as a probability of each attempt turning positive(e.g. recommend products) while $1-q$, turning negative(otherwise defeats or avoidance). At each step t, a positively activated node v will try to activate each of its non-active neighbours positively, and if successful (with a success probability), the neighbour then becomes active in next step $t+1$, but it only turns positive with probability q or negative with probability $1-q$. Similarly, a negatively activated node v' will also try to negatively activate its non-active neighbours with a success probability, and if successful the neighbours become negative. If several nodes(both positive and negative) try to activate the same node in one step, then the order of activation attempts is random.

B. Threshold Models

A Threshold Model is a concept from mathematics or statistics. In a threshold model, a threshold value will be set to distinguish ranges of values where the behaviour predicted by the model varies in some important ways. The threshold models are firstly proposed by Mark Granovetter [8]. In the work of [8], they use this model to study collective behaviour,

which aimed at treating binary decisions problems, such as diffusion of innovations, spreading rumours and diseases, voting and so on. The distribution of the thresholds determines the outcome of the aggregate behaviour (e.g. voting, opinions). In other words, this threshold represents the number of other agents in the population or local neighbourhood following certain particular activity. Each agent has a threshold that, when exceeded, leads the agent to adopt an activity [32]. In threshold, a node will change status(e.g. become active) if the fraction of its neighbours exceeds the threshold set before. This model can avoid sudden change which might not really reflect the behaviour in real life.

*1) **Linear Threshold**:* Linear threshold(LT) is proposed in [8] and used to study IM problem in [3] [35] [36]. In linear threshold model, denote $N(v)$ as the neighbours of node v, θ_v as thresholds for every node $v \in V$. For every neighbour $u \in N(v)$, $\overline{uv}$ has a nonnegative weight $w_{u,v}$ which is subject to $w_{u,v} \leq 1$ and $\sum_{u \in N} w_{u,v} \leq 1$. Beginning from seeds, at each step t, a not-yet-activated node will become active at next step $t+1$ if

$$\sum_{u \in N^a(v)} w_{u,v} \geq \theta_v \tag{5}$$

where $N^a(v)$ denotes the set of active neighbours of node v. The diffusion process of this model is deterministic and in the work of Lu et al. [26], they prove the NP-hard of IM under LT model referred as following:

***Definition** 4:* There is no $n^{1-\epsilon}$ polynomial time approximation for IM problem unless $P = NP$.

Proof: For a finite set S, $|S|$ is the number of elements in S, the input of a set cover problem is $S_1, S_2, ..., S_m$, and S, where $S_1, S_2, ..., S_m$ are m subsets of set S. The target is to find l subsets $S_{i1}, S_{i2}, ..., S_{il}$ such that $S_{i1} \cup S_{i2} \cup ... \cup S_{il} = S$. Given a set cover problem, reduce it to the influence maximization problem as follows:

1) For each subset S_i, create a node v_i. Then there are totally m nodes for all subsets.
2) Let $n = |S|$, and assume $S = \{a_1, a_2, ..., a_n\}$. Create a node u_j for each element $a_j \in S$. Then there are n nodes for all elements.
3) Let c denote an arbitrary constant, create additional $(m+n)^c$ nodes $w_1, w_2, ..., w_{(n+m)^c}$.
4) If subset S_i contains elements a_j, create a directed edge $e_{i,j}$ from v_i to u_j, with weight $b_{j,i} = 1$.
5) For each node u_j , create $(n+m)^c$ edges from u_j to all the nodes $w_1, w_2, ..., w_{(n+m)^c}$, with weight $b_{k,j} = \frac{1}{n}$.

Assume all the threshold $\theta_i = 1$ for any nodes v_i and $n > m$:

- if the set cover problem has a solution of l subsets, then the influence maximization problem has a solution with $f(A) = m + n + (m+n)^c$, where $|A| = l$.
- if the set cover problem has no solution, then the influence maximization problem for any initial seed set $|A| = l$, has final active set $f(A) \leq n + m$.

Therefore, an $n^{1-\epsilon}$-factor polynomial time approximation for the influence maximization problem implies $P = NP$. ■

*2) **Majority Threshold**:* Majority threshold(MT) model, different the general linear threshold model, a not-yet-activated node $v \in V$ with a threshold $\theta_v = \frac{1}{2}d(v)$ becomes active when the majority of its neighbours are active. This model is widely studied in voting systems, distributing computing(e.g. [37]). It has been proven the same hardness with general threshold model in the work of Ning Chen [38].

*3) **Small Threshold**:* Small threshold(ST) model is a model in which all thresholds are small constants. Then the IM problem can be easily solved by selecting an arbitrary node in each connected component. However, it can be showed by Ning Chen [38] that either $\theta = 1$ or $\theta = 2$ for this models has no changeable for the hardness. Later Dreyer [39] proves that if the threshold of any node is θ_v for any $\theta \geq 3$, the problem is still NP-hard.

*4) **Unanimous Threshold**:* Unanimous Threshold (UT) model has the threshold for each node $\theta_v = d(v)$, which is equal to its degree. It is the most influence-resistant model among all the threshold models and usually used in studying complex network security and vulnerability. An example to explain is that in an ideal network with virus exists, a computer will be infected if all its neighbours are being infected. This can be considered a special case and turns to be the Vertex Cover problem. Therefore, under this model, IM problem is still NP-hard.

C. Equivalence of Cascade Models and Threshold Models

In the last of paper of Kempe [3], the equivalence of IC models and LT models are proven by using more general frameworks of IC models and LT models.

- **Generalized cascade model**. Compared with the specific cascade models, if allows the probability that u successfully activates its neighbour v to depend on the other active neighbours of v that have tried, a generalized cascade model can be generated. Change the activation probability $P_{u,v}$ to an incremental function $p_v(u, S)$, where $\{u\}$ and S are two disjoint subsets of $N(v)$. In each step, the process is the same as IC model: when a newly activated node u attempts to activate a not-yet-activated node v, it succeeds with probability $p_v(u, S)$, where S denotes the set of nodes that have already made their attempts. Then the IC model can be considered as a special case of the generalized cascade model, in which $p_v(u, S)$ is set to a constant $p_{u,v}$ and independent of S. Besides, the order-independence which has been introduced in the IC model still holds here.
- **Generalized threshold model**. In the general threshold model, each node v has a threshold θ_v, and associates with a function f_v that maps the set of its neighbours $N(v)$ to the range $[0, 1]$ and subject to the condition $f_v(\emptyset) = 0$. This function could be an arbitrary monotone function f_v. The dynamic of diffusion process follows the general structure of the LT model. However, a node v becomes active at step $t+1$ if and only if $f_v(N^a(v)) \geq \theta_v$, where $N^a(v)$ is the subset of active neighbours of v at step t. Thus, the LT model is a special

case for the generalized threshold model, in which the threshold function is subject to $f_v = \sum_{u \in N^a(v)} w_{u,v}$ and $\sum_{u \in N(v)} w_{u,v} \leq 1$.

***Definition** 5 (Equivalence of Cascade Models and Threshold Models):* If the threshold function f_v is chosen independently and uniformly at random, then these two generalized models are equivalent.

Proof: Let f_v be a threshold function of a general threshold model, and S be the set of nodes that have already tried to activate v. Then in order to define an equivalent cascade model, the probability of additional node u can activate v if all the nodes in S have failed needs to be known. Once the node in S failed, node $v's$ threshold θ_v should be in the range $(f_v(S), 1]$. Therefore, with the constraint that it should be uniformly distributed, the probability that a neighbour $u \notin S$ successfully activate v is

$$p_v(u, S) = \frac{f_v(S \cup \{v\}) - f_v(S)}{1 - f_v(S)} \tag{6}$$

where nodes in S failed to activate v. It is easy to see that the generalized cascade model can be converted to the generalized threshold model with this function.

On the other side, let v be a node in the cascade model, with its neighbour set denoted by $N(v) = \{u_1, u_2, ..., u_k\}$. All the nodes in $N(v)$ have tried to activate v in an order T and if assume $T = \{u_1, u_2, ..., u_k\}$, and $N(v)_i = \{u_1, u_2, ...u_i\}$, then the probability that v hasn't been influenced is $\prod_{i=1}^{k}(1 - p_v(u_i, N(v)_{i-1}))$. According to the order-independence, this value is not affected by the order of u_i, but only depends on the set $N(v)$, then there has

$$f_v(S) = 1 - \prod_{i=1}^{k}(1 - p_v(u_i, N(v)_{i-1})) \tag{7}$$

Then, the equivalence of cascade models and threshold models is proven. ■

D. Other Models

*1) **Competitive Influence Diffusion Models**:* In the work of Tim Carnes et al. [40], they consider competing situation for marketing products which are more interesting than transitional case. For only two competing products(e.g. technology A or B), there are two assumptions, one is that the consumers use only one of the two products and will influence their friends on their decision on which product to use, and the other is that the competitors has a fixed budget available that can be used to target only a subset of consumers. In [40], they propose two models to describe how two technologies simultaneously diffuse over a given network.

- *Distance-based Model*. A distance-based model take the location of a node v in the network into consideration and regard that a node will mimic the behaviour(e.g. might be a consumer) of an early adopter if their distance in the social network is relatively small. In [40], they propose that the expected number of nodes which adopt A will be denoted by

$$\rho(I_A|I_B) = \mathbb{E}\left[\sum_{u \in V} \frac{v_u(I_A, d_u(I, E_a))}{v_u(I_A, d_u(I, E_a)) + v_u(I_B, d_u(I, E_a))}\right] \tag{8}$$

where the expectation is over the set of active edges. I_A and I_B are the initial sets of adopters of A and B respectively, and I is their union set. $d_u(I, E_a)$ denotes the shortest distance from u to I along the edges in E_a. After fixing I_B and trying to determine I_A so as to maximize the expected number of nodes that adopt technology A would be:

$$max\{\rho(I_A|I_B) : I_A \subseteq (V - I_B), |I_A| = k\} \tag{9}$$

- *Wave Propagation Model*. In this model, the diffusion propagation is happened in discrete steps. In step t, all nodes that are at distance at most $t1$ from some node in the initial sets(seeds) have adopted technology A or B, and all nodes for which the closest initial node is farther than $t1$ do not have a technology yet (where the distance is again with respect to active edges). The nodes at a distance t from the initial sets now choose one of their neighbours that are at distance $t1$ independently at random, and adopt the same technology as this neighbour. Then

$$P(u|I_A, I_B, E_a) = \frac{\sum_{v \in N(v)} P(v|I_A, I_B, E_a)}{|N(v)|} \tag{10}$$

where $P(v|I_A, I_B, E_a)$ is the probability that node v adopts technology A when the initial sets for technologies A and B are I_A and I_B, respectively, the set of active edges is E_a, u is a node for which the closest node in $I = I_A \cup I_B$ is at distance t, $N(v)$ is the set of neighbours of u that are at distance $t-1$ from I, where all distances are again with respect to active edges. For initial set I_A, I_B, let

$$\pi(I_A|I_B) = \mathbb{E}\left[\sum_{v \in V} P(v|I_A, I_B, E_a)\right] \tag{11}$$

denote the expected number of nodes that adopt technology A. For fixed I_B, a solution is

$$max\{\pi(I_A|I_B) : I_A \subseteq (V - I_B), |I_A| = k\} \tag{12}$$

*2) **Weight-proportional Threshold Model**:* In the work of Borodin et al. [41], a weighted-proportional threshold(WT) model is proposed. In step t, let Φ^t denote the set of active nodes, then for the sets of A-active and B-active, the notation is Φ_A^t and Φ_B^t separately. Given two different seeds S_A and S_B at the beginning, in each step, every inactive node v changes its status according to the incoming influence from its currently active neighbours as follows: v becomes active when $\sum_{u \in \Phi^t} w_{u,v} \geq \theta_v$ is satisfied. In addition, v becomes a A-active node with probability

$$Pro\left[v \in \Phi_A^t | v \in \Phi^t \backslash \Phi^{t-1}\right] = \frac{\sum_{u \in \Phi_A^t} w_{u,v}}{\sum_{u \in \Phi^t} w_{u,v}} \tag{13}$$

Otherwise, it adopts cascade B. Intuitively, by adding one more node to the initial set S_A, the spread of cascade A could be expended, however it is not true since the influence function $\sigma(\cdot)$ is neither monotone nor submodular under this model which can be proven by a count example showed in [41].

*3) **Separated Threshold Model**:* Compared to previous WT model, a node might have different threshold towards different competitors. Then separated threshold(SepT) model is proposed to model this case in [41]. Consider two thresholds θ_v^A and θ_v^B for a node v towards competitors A and B and each edge $\overline{uv} \in E$ associated with two weights $w_{u,v}^A$ and $w_{u,v}^B$ corresponding to A and B, respectively. Both thresholds are subject to the constraints as in the LT model. In each step t, every not-yet-activated node v can become A-active when $\sum_{u \in N^a(v) \cap \Phi^{t-1}} w_{u,v}^A \geq \theta_v^A$ or be B-active when $\sum_{u \in N^a(v) \cap \Phi_B^{t-1}} w_{u,v}^B \geq \theta_v^B$, or adopts a cascade uniformly at random when both thresholds are exceeded. In this model, the influence function $\sigma(\cdot)$, however, is monotone but not submodular which is also proven in [41] by a counterexample.

*4) **Latency-aware Independent Cascade Model**:* In the work of Liu et al. [28], they proposed a new independent cascade model with latency on influence being considered, called latency-aware cascade model(LAIC) model and they use this model to study the time constraint influence maximization problem. In this model, an inactive node u will be attempted to activate by its active neighbour v in step $t+\delta_t$ with probability $P_{u,v}P_u^{lat}(\delta_t)$, where δ_t denotes the influencing delay and is randomly generated from the delay distribution P_u^{lat}. Note that a node can only be activated once and be activated at the earliest activation time while the rest successful activations will be ignored. This diffusion process will also terminate when no new nodes activated.

It is proven that the IM problem remains NP-hard in the paper of [28].

*5) **Topic-aware Cascade Model**:* Topic aware models are proposed by Barbieri et al. [42] and they extend the IC and LT model with topic aware analysis included.

- *Topic-aware Independent Cascade Model (TIC)*. In topic-aware IC model, the propagation probability depends on the topic. For each edge $\overline{uv} \in E$ and each topic $z \in [1, K]$, there is a probability $p_{u,v}^z$ denoting the strength of the influence exerted by node v on node u on topic z. Besides, for each item i, use $\gamma_i^z = P(Z = z|i)$ subject to $\sum_{z=1}^K \gamma_i^z = 1$ to denote a distribution over the topics, which is for each topic $z \in [1, K]$.
  This model is extended from IC model and therefore the diffusion process happens like that in IC model: a current active node v at step t has an attempt to activate a nonactive neighbour u independently of history so far with a success probability that is the weighted average of the link probability w.r.t. the topic distribution of the item i:

$$p_{u,v}^i = \sum_{z=1}^{K} \gamma_i^z p_{u,v}^z \tag{14}$$

- *Topic-aware Linear Threshold Model (TLT)*. Like LT model, with the same notation in TIC model but the sum of incoming weights in each node and for each topic is no more than 1. There is also a threshold θ_u uniformly at random from $[0, 1]$ for each node u. At step t, a not-yet-activated node u is submitted to an influence weight

$$W_i^t(u) = \sum_{z=1}^{K} \sum_{u \in \mathcal{F}_i(u,t)} \gamma_i^z p_{v,u}^z \tag{15}$$

  where $\mathcal{F}_i(u, t)$ denotes the set of nodes that have a link to u and that at step t have already adopted the item i. If $W_i^t(u) \geq \theta_u$, then u will change to active at next step $t + 1$.
  In the work of [42], it has been proven that these two extended models are still monotone and submodular.

*6) **Voter Model**:* Voter model is proposed by Even-Dar in [43]. In this model, denote $G = (V, E)$ as the social network which is an undirected graph with self loops and $N(v)$ as the set of neighbours of node v. The diffusion process starts from an arbitrary initial $0/1$ assignment to a node of G(seed), then at each step t, each assigned node can uniformly at random select one of its neighbour $u \in N(v)$ and adopts its opinion(0 or 1). More formally, this process can be represent as following: starting from any assignment $f_0 : V \to \{0, 1\}$, then at next step $t + 1$:

$$f_{t+1}(v) = \begin{cases} 1, & \text{with probability } \frac{|\{u \in N(v) : f_t(u) = 1\}|}{|N(v)|} \\ 0, & \text{with probability } \frac{|\{u \in N(v) : f_t(u) = 0\}|}{|N(v)|} \end{cases} \tag{16}$$

Note that the voter model is a random process whose behaviour depends on the initial assignment f_0 and ideally the later assignment $f_t(v) = 1$ indicates whether v is active(might be using certain product). This model can naturally be expected maximized with all nodes assigned 1, however, it is usually a limit on the budget for such diffusion propagation in real life. This model is usually be used for variant IM problem with budget constraints.

*7) **Ising Model**:* In the work of Liu et al. [9], they consider to specify the correlations between different users on a social network and propose the ising model. In this ising model, the opinion of node v is defined as O_v with $O_v = +1$ if the node v is in favour of the subject otherwise $O_v = -1$. They denote by $\mathbf{O} = \{O_1, O_2, ..., O_n\}$ and $\mathbf{o}$ to be realization of $\mathbf{O}$.

$$O_u = F_u\left(\{O_v\}_{v \in N(u)}\right) \tag{17}$$

Then, they assume the network contains nodes with pre-determined opinions whose opinions are independent of their neighbours' opinions. Name a node a *positive seed* if it holds a pre-determined positive opinion($O_v = +1$) and denote the subset of positive seeds by Ψ^+. For negative nodes with negative opinions($O_v = -1$), also name them *negative seeds* and denote the set of them by Ψ^-. Then to compute the

expected number of two opinions on a network, follows two equations:

$$N^+(\Psi^+, \Psi^-) = \mathbb{E}\left[\sum_{v\in V} 1_{O_v=+1}|\Psi^+, \Psi^-\right] \tag{18}$$

$$N^-(\Psi^+, \Psi^-) = \mathbb{E}\left[\sum_{v\in V} 1_{O_v=-1}|\Psi^+, \Psi^-\right] \tag{19}$$

And the randomness arises due to probabilistic correlations between opinions of neighbours. So, under this model, the IM problem is to find the solution:

$$\max_{\tilde{\Psi}^+:|\tilde{\Psi}^+|=m} N^+(\tilde{\Psi}^+, \Psi^+, \Psi^-) = \mathbb{E}\left[\sum_{v\in V} 1_{O_v=+1}|\tilde{\Psi}^+, \Psi^+, \Psi^-\right] \tag{20}$$

where m is the number of the targeted users selected to express positive opinion.

To find the solution, a Markovian random field is defined and a placement algorithm is designed for this model.

The probability of opinion formed $\mathbf{o}$ is assumed to be

$$Pro(\mathbf{o}) = \frac{1}{\mathbf{Z}} \exp\left(\sum_{\overline{uv}\in E} \frac{W_{\overline{uv}}}{\mathbf{T}} o_u o_v\right) \tag{21}$$

where $W_{\overline{uv}}$ is a parameter indicating the correlation between u and v, $\mathbf{T}$ is a parameter that indicates the time remaining for a decision to be made. When $\mathbf{T} = 0$ opinions get fixed, and $\mathbf{Z} = \sum_{\mathbf{o}} \exp\left(\sum_{\overline{uv}\in E} \frac{W_{\overline{uv}}}{\mathbf{T}} o_u o_v\right)$ is the normalizing factor. Therefore,

$$Pro(o_u|\mathbf{o}\backslash\{o_u\}) = \frac{\exp\left(\sum_{v:\overline{uv}\in E} \frac{W_{\overline{uv}} o_u o_v}{\mathbf{T}}\right)}{\exp\left(\sum_{v:\overline{uv}\in E} \frac{W_{\overline{uv}} o_v}{\mathbf{T}}\right) + \exp\left(-\sum_{v:\overline{uv}\in E} \frac{W_{\overline{uv}} o_v}{\mathbf{T}}\right)} \tag{22}$$

So, This graphical model is a Markovian random field (MRF).

E. Summary

This part introduces some models studied in IM problem but does not cover all of them. Beyond the models mentioned before, there are other models such as, a data-based model *credit distribution* of Goyal et al. [44], a model on influence transitivity of Xu et al. [20], an expectation model of Lee et al. [24], conformity-aware cascade models of Li et al. [45] and Li et al. [46], a containment model of Peng et al. [47], Three Steps Cascade Model of Qin et al. [48], an extended voter model of Li et al. [49], an extended model of Zhu et al. [50], an extended threshold model of Lu et al. [22], etc.

IV. Techniques

This section will talk about the techniques(e.g. algorithms, frameworks, strategies, approaches) on IM problems. Techniques applied to IM problems are important since analysis on large-scale social networks is consumable. As the analysis of complexity in Sec. II, this problem is #P-hard. While even under certain models, it is still NP-hard, so, there are no deterministic techniques which can give a feasible solution in polynomial time. The original greedy algorithm, though, can give the best solution so far but it is not feasible if applied to large-scale networks. Thus, heuristic methods are widely studied since it can give a matchable solution compared to greedy strategy. There are some other techniques like learning methods, community-based techniques, etc.

A. Greedy Techniques

Greedy strategies are the most approximate algorithm that also acts as the benchmark of IM problems. It is considered in [51] and then many modifications on the general greedy algorithms are being studied which leads to some efficient greedy approaches.

*1) **General Greedy Algorithm**:* In the work of Domingos and Richardson [51], they consider a general hill-climbing algorithm which is always within a factor of $(1 - 1/e)$ of the optimal for IM problem. The algorithm can be referred as Algorithm 1.

Algorithm 1 GeneralGreedy(G, k)

Input: a network $G = (V, E)$, the number of seeds k
Output: a seed set S

```
S ← ∅, R ← 20000
for i = 1 to k do
    for each vertex v ∈ V\S do
        s_v ← 0
        for i = 1 to R do
            s_v+ = |RanCas(S∪{v})| //RanCas is the access model
        end for
        s_v ← s_v/R
    end for
    S ← S ∪ {argmax_{v∈V\S}{s_v}}
end for
return S
```

In this algorithm, access every node each iteration after selecting a node as seed, then select the node with maximal reward as new seed. The time complexity of this algorithm is $O(knRm)$ since $RanCas(S)$ takes $O(m)$ time.

*2) **NewGreedy Algorithm**:* Since the general greedy algorithm cannot handle large netwoks, then in [10], they proposed two optimized greedy algorithms under IC and WC models. These two model-specific algorithms use a generated graph G' to cut down the computation of IM problem and they can be referred as Algorithm 2 and Algorithm 3.

Algorithm 2 NewGreedyIC(G, k)

Input: a network $G = (V, E)$, the number of seeds k
Output: a seed set S

$S \leftarrow \emptyset$, $R \leftarrow 20000$
for $i = 1$ to k **do**
  Set $s_v \leftarrow 0$ for all $v \in V \backslash S$
  for $i = 1$ to R **do**
    Compute G' by removing each edge from G with probability $1 - p$
    Compute $R_{G'}(S)$
    Compute $|R_{G'}(\{v\})|$ for all $v \in V$
    for each node $v \in V \backslash S$ **do**
      if $v \notin R_{G'}(S)$ **then**
        $s_v += |R_{G'}(\{v\})|$
      end if
    end for
  end for
  Set $s_v \leftarrow s_v / R$ for all $v \in V \backslash S$
  $S \leftarrow S \cup \{\text{argmax}_{v \in V \backslash S}\{s_v\}\}$
end for
return S

This modified greedy algorithm is based on IC model which use a randomly generated graph G' to delete some edges to firstly determine the selected edges for propagation. Either BFS or DFS can be used to generate the G'. The running time for this algorithm is $O(kRm)$.

Algorithm 3 NewGreedyWC(G, k)

Input: a network $G = (V, E)$, the number of seeds k
Output: a seed set S

$S \leftarrow \emptyset$, $R \leftarrow 20000$, $T \leftarrow 5$
for $i = 0$ to k **do**
  Initialize $s_v \leftarrow 0$ for all $v \in V$
  for $j = 1$ to R **do**
    $G' \leftarrow RanWC(G)$
    Compute DAG G'^* and weights $w(v^*)$ for all $v^* \in V^*$
    for $\ell = 1$ to T **do**
      for each $v^* \in V^*$ **do**
        $s^{\ell}_{v^*} \leftarrow 0$
        Generate random value $X^{\ell}_{v^*}$ from the exponential distribution with mean $1/w(v^*)$
        Compute $Y^{\ell}_{v^*} \leftarrow \min_{u^* \in R_{G'^*}(S^* \cup \{v^*\})} X^{\ell}_{u^*}$
        $s^{\ell}_{v^*} += Y^{\ell}_{v^*}$
      end for
      For each $v \in V \backslash S$, $s_v += (T - 1)/s^{\ell}_{v^*}$
    end for
  end for
  Set $s_v \leftarrow s_v / R$ for all $v \in V \backslash S$
  $S \leftarrow S \cup \{\text{argmax}_{v \in V \backslash S}\{s_v\}\}$
end for
return S

This algorithm is used under the WC model. Different the IC model, the graph is a directed graph G', so, the process to obtain G' should be redesign and the authors of [10] et al. adopt the randomized algorithm from Cohen's work [52]. The randomized algorithm will construct a DAG by computing all strongly connected components of G' and collapsing each strongly connected component into one node with weight being the size of the strongly connected component. The time complexity of this algorithm is $O(kRTm)$.

*3) **State Machine Greedy Algorithm**:* In the work of Heidari et al. [53], they present a fast greedy algorithm called *State Machine Greedy* by reducing the computation into two parts: counting the traversing nodes in the propagation and Monte-Carlo graph construction in the simulation of diffusion. The algorithm is described in two flowcharts referred as Fig. 3 and Fig. 4 in **Appendix**. In the flowchart of SMG, a single state machine saves calculated propagation of i nodes of S in all Monte-Carlo graphs as the last state and uses it to prevent recalculation of this propagation in the next step. The state machine will find R graphs generated in the Monte-Carlo simulation with inactive nodes so as the one with maximum marginal gain. The most influential nodes will be added to S and the process will repeat to find the next one. For the function called by SMG, F, its flowchart also provide details. The function F only calculates the increased propagation of w instead of calculating propagation for $i+1$ elements in $S \cup \{w\}$. The complexity of this algorithm cannot be shown by time complexity, instead, the authors provide two tables containing the *Complexity of Trigger Nodes* and the *Complexity of Graph Construction*.

B. Lazy-Forward Techniques

Based on a lazy-forward optimization, cost-effective lazy forward algorithms and its extended algorithms can improve the quadratic nature of the greedy algorithm. These algorithms also combine the submodularity of this problem proven in Sec. II.

*1) **CELF Selection Algorithm**:* The notable work for improving the greedy algorithm is done by Leskovec et al. [54]. The core idea of this algorithm is that the marginal gain of a node selected in the current iteration cannot be better than its marginal gain in the previous iterations. CELF maintains a table $\langle v, \Delta_v(S) \rangle$ sorted on $\Delta_v(S)$ in descending order where S denotes the current seed set and $\Delta_v(S)$ denotes the marginal gain of node v w.r.t S. The $\Delta_v(S)$ will be re-evaluated only for the top node at a time and if needed, the table will resort. Then a node will be selected as new seed if it remains at the top. This algorithm can be referred as the combination of Algorithm 4 and Algorithm 5.

Algorithm 4 LazyForward(G, k, $type$)

Input: a network $G = (V, E)$, the number of seeds k, compute type $type$
Output: a seed set S

$S \leftarrow \emptyset$
while $\exists v \in V \backslash S : c(S \cup \{v\}) \leq B$ **do**
  for each node $v \in V \backslash S$ **do**
    $cur_v \leftarrow false$
  end for
  while true **do**
    if $type = UC$ **then**
      $v^* \leftarrow \text{argmax}_{v \in V \backslash S, c(S \cup \{v\}) \leq B} \delta_v$
    else if $type = CB$ **then**
      $v^* \leftarrow \text{argmax}_{v \in V \backslash S, c(S \cup \{v\}) \leq B} \frac{\delta_v}{c(v)}$
    end if
    if cur_v **then**
      $S \leftarrow S \cup \{v^*\}$
      break
    else
      $\delta_v \leftarrow R(S, v) - R(S)$
      $cur_v \leftarrow true$
    end if
  end while
end while
return S

The algorithm 4 can compute the marginal gain and return the seed set S for appointed type. The R denotes the penalty reductions introduced in [54].

Algorithm 5 CELF(G, k)

Input: a network $G = (V, E)$, the number of seeds k
Output: a seed set S

$S_{UC} \leftarrow LazyForward(G, k, UC)$
$S_{CB} \leftarrow LazyForward(G, k, BC)$
return agrmax$\{R(S_{UC}), R(S_{CB})\}$ //return the set with maximal result

Algorithm 5 considers the maximal result for Algorithm 4 under *UC* and *CB* then choose the nodes yielding the result.

*2) **CELF++ Algorithm**:* After the work of Leskovec et al on *CELF* algorithm, Goyal et al. propose an improved CELF algorithm, called *CELF++* in [36]. In this algorithm, $\sigma(S)$ denotes the spread of seed set S and a heap Q is maintained with nodes corresponding to users in the network G. Each node in Q corresponding to user u stores a tuple of the form $\langle u.mg1, u.prev\_best, u.mg2, u.flag \rangle$. In the tuple, $u.mg1 = \Delta_u(S)$, which is the marginal gain of u w.r.t. the current seed set S; $u.prev\_best$ is the node whose marginal gain is the maximum among all the users examined in the current iteration; the third element $u.mg2 = \Delta_u(S \cup \{prev\_best\})$ and the last element $u.flag$ denotes the iteration number when $u.mg1$ was last updated. The idea of this algorithm is that if the value $u.prev\_bset$ of a node u is picked as a seed in the current iteration, no recomputing on the marginal gain of the node u is needed. The process of this algorithm can be seen in Algorithm 6.

Algorithm 6 CELF++(G, k)

Input: a network $G = (V, E)$, the number of seeds k
Output: a seed set S

$S \leftarrow \emptyset$, $Q \leftarrow \emptyset$, $last\_seed \leftarrow null$, $cur\_best \leftarrow null$
for each $u \in V$ **do**
  $u.mg1 \leftarrow \sigma(\{u\})$, $u.prev\_best \leftarrow cur\_best$
  $u.mg2 \leftarrow \sigma(\{u, cur\_best\})$, $u.flag \leftarrow 0$
  Add u to Q
  Update cur_best based on $mg1$.
end for
while $|S| < k$ **do**
  $u \leftarrow$ top (root) element in Q
  if $u.flag = |S|$ **then**
    $S \leftarrow S \cup \{u\}$, $Q \leftarrow Q - \{u\}$, $last\_seed \leftarrow u$
    continue
  else if $u.prev\_best = last\_seed$ **then**
    $u.mg1 \leftarrow u.mg2$
  else
    $u.mg1 \leftarrow \Delta_u(S)$, $u.prev\_best = cur\_best$
    $u.mg2 \leftarrow \Delta_u(S \cup \{cur\_best\})$
  end if
  $u.flag \leftarrow |S|$
  Update cur_best
end while
return S

C. Metaheuristic Techniques

Beyond greedy or optimized greedy algorithms, metaheuristic algorithms are being used to study the IM problem: *Simulated Annealing* [55] and *Ant Colony Optimization* [56].

*1) **Simulated Annealing**:* In the work of Jiang et al. [55], they apply simulated annealing(SA) to the IM problem. The fitness function $\sigma(S)$ is defined under the IC model where $S \subset V$. The algorithm can be seen in Algorithm 7.

Algorithm 7 SA based Top-k mining algorithm

Input: a network $G = (V, E)$, the number of seeds k, the initial temperature T_0, termination temperature T_f, the number of inner loop q, the amount to cut down the current temperature in the outer loop ΔT
Output: a seed set S

$t \leftarrow 0$, $T_t \leftarrow T_0$, $count \leftarrow 0$
Randomly select an initial seed set $S \subset V$ subject to $|S| = k$
while $T_t < T_f$ **do**
  Calculate $\sigma(S)$
  $S' \leftarrow F(S, G)$, {Create a neighbour solution set}
  $count \leftarrow count + 1$
  calculate the change of the fitness $\Delta f \leftarrow \sigma(S') - \sigma(S)$
  if $\Delta f > 0$ **then**
    $S \leftarrow S'$
  else
    Create a random number $\xi \in U(0, 1)$
    if $exp\left(\frac{\Delta f}{T_t}\right) > \xi$ **then**
      $S \leftarrow S'$
    end if
  end if
  if $count > q$ **then**
    $T_t \leftarrow T_t - \Delta T$, $t \leftarrow t + 1$, $count \leftarrow 0$
  end if
end while
return S

And to speed up the calculations on $\sigma(S)$, they also propose an Expected Diffusion Value(EDV) to replace the diffusion simulations and develop the SAEDV algorithm. However, this change might cause the solution to be inaccuracy, so, they then propose SASH algorithm to enhance the accuracy of the solution.

D. Heuristic Techniques

In some work of the IM problem on large-scale social networks, heuristic techniques are being considered. Although in most cases the solution obtained by heuristic algorithms is worse than that obtained by greedy or lazy-forward-based greedy algorithms, with the specific design under the certain model, the heuristic algorithms can make the analysis on large networks feasible.

1) ***DegreeDiscount****:* In the work of Chen [10], except the improved greedy algorithms, they also propose a heuristic algorithm named DegreeDiscount algorithm which can run in $O(k\log n + m)$ if combined with a Fibonacci heap. They apply this algorithm under IC model and the algorithm can be referred in Algorithm 8.

Algorithm 8 DegreeDiscount(G, k)

Input: a network $G = (V, E)$, the size of seeds k
Output: a seed set S

$S \leftarrow \emptyset$
for each vertex $v \in V$ **do**
  calculate its degree d_v
  $dd_v \leftarrow d_v$
  $t_v \leftarrow 0$
end for
for $i = 1$ to k **do**
  $u \leftarrow \text{argmax}\{dd_v | v \in V \backslash S\}$
  $S \leftarrow S \cup \{v\}$
  for each neighbour v of u **and** $v \in V \backslash S$ **do**
    $t_v \leftarrow t_v + 1$
    $dd_v \leftarrow d_v - 2t_v = (d_v - t_v)t_v p$
  end for
end for
return S

In this algorithm, let v be a neighbour of vertex u. If u has been selected as a seed, then when considering selecting v as a new seed based on its degree, the edge vu towards its degree should not be counted. Thus the algorithm discount v's degree by one due to the presence of u in the seed set, and it also does the same discount on v's degree for every neighbour of v that is already in the seed set. [10] This algorithm conducts a discount on the degree so as to optimize the running time with performance much faster than the greedy method.

E. SIMPATH

SIMPATH algorithm is proposed by Goyal et al.in [57], which is an efficient and effective algorithm for IM problem under the LT model. This algorithm use *simpath-spread* which shows that the spread of the influence from a node can be computed by summing the weights (e.g. probabilities) of all simple paths originating from it. Then combining with CELF, the resulting algorithm can apply to IM problems.

F. Learning-based

With the development of machine learning, some learning-based algorithms are developed to apply to IM problem and outperform the state-of-art algorithms.

In the work of Mohammad et al [58], they firstly applied a **learning automation based** algorithm to solve the minimum positive influence dominating set (MPIDS) problem, and then further use the MPIDS for IM problem.

In [11], supervised learning is applied to IM problem. They propose an innovative **Transfer Influence Learning**(TIL) based on three real networks. To maximize the influence propagation, a greedy algorithm should be run at first to help the proposed framework completely avoid the Monte Carlo simulation and then the algorithm trains a classifier based on the results of the greedy algorithm and uses it to directly decide whether a node should be selected. The framework can be referred in Fig. 1.

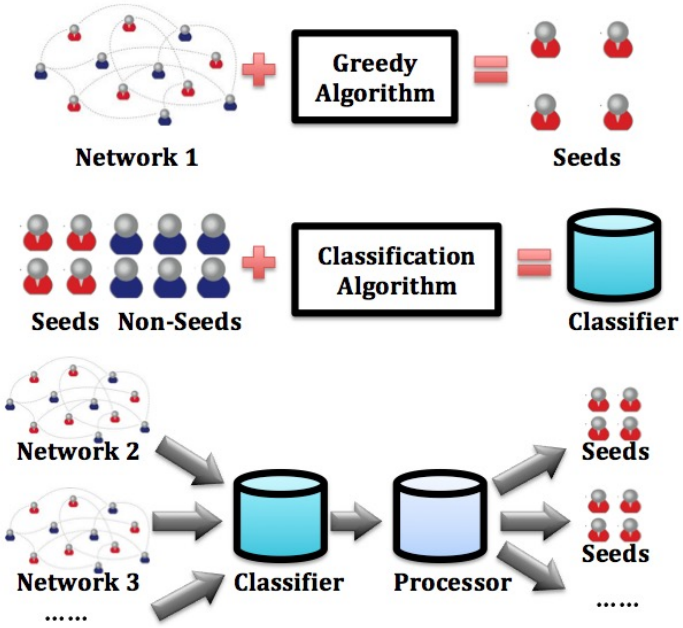

Fig. 1. Transfer Influence Learning (TIL) Framework

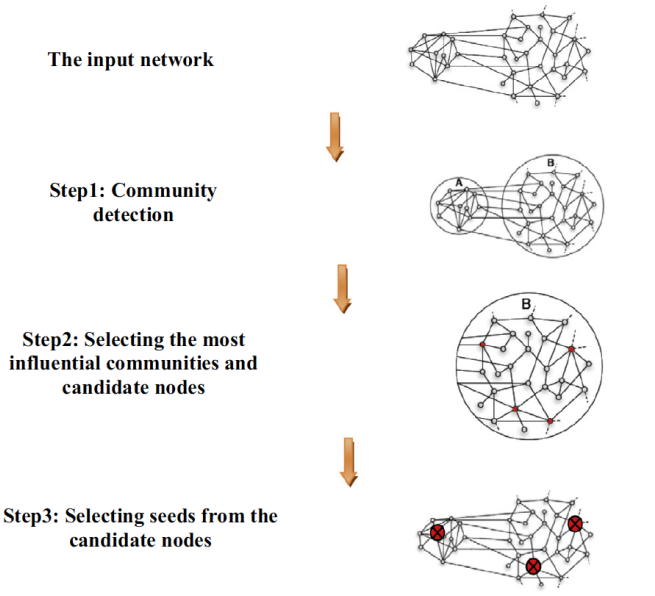

Fig. 2. The main steps of the proposed algorithm: In step1, the input network is reduced into a new small network by use of a community detection algorithm, in step 2, some candidate nodes are selected from each community and in step 3, final seeds are selected from the candidate nodes.

G. Community based

Different from the algorithm mentioned before, some algorithms consider the network from the structure. Community-based algorithms are studied in [35] [12] [59] [60] [61]. These algorithms will detect community on social networks and solve the IM through the structure of the input network data.

For an example, in the work of Rahimkhani [35], the algorithm will detect the community on a social network and then form a new network according to the structure. In the new network, each node represents a community. Then the most central nodes of the new network are detected based on the betweenness centrality measure and each node of the new network contains the nodes of its corresponding community. Thus, proportional to the size of the community, a number of nodes are selected from each community based on the degree centrality to form the candidates set. Then top-k most influential nodes are chosen from the candidates set based on the paths that exist in the network. The process can be divided into 3 steps shown as Fig. 2.

H. Summary

In addition to these techniques, there are other techniques developed to study IM problems like ranking-based techniques in [62] [13], a divide-and-conquery strategy in [65], a dynamic programming approach in [66], coritivity-based algorithms in [64] and a potential-based algorithm in [63], an UCB-based algorithm in [21], a centrality-measures-based algorithm in [67], etc.

V. Datasets

This section will introduce some widely used datasets on IM problems.

A. Collaboration Networks

1) ca-HepTh: ca-HepTh [70] is a collection of the authors of the papers in "High Energy PhysicsTheory" obtained from Arxiv. This dataset is an undirected graph with 9877 nodes and 25998 edges.

2) GR-QC: GR-QC [69] is a collection of the authors of the papers in "General Relativity and Quantum Cosmology Category". This dataset is also an undirected graph containing 5242 nodes and 14496 edges.

3) DBLP: DBLP [71] is a collaboration network obtained from the DBLP Computer Science Bibliography Database. This network has $655K$ nodes and $2.0M$ edges which is a large-scale undirected graph.

B. Social Networks

1) Facebook Network: Facebook network [5] is a network with social circles from the popular website facebook.com. This network is an undirected graph including 4039 nodes and 88234 edges.

2) Flixster Network: Flixster network [68] is a directed graph with $1M$ nodes and $28M$ edges obtained from Flixster(www.flixster.com) which is one of the main players in the mobile and social movie rating business.

C. Summary

Table II shows the summary of the dataset introduced in this section. Datasets of IM problems can be simply divided into directed and undirected graph by edge direction. If divided by the size, there are small-size networks and large-scale networks.

TABLE II
SUMMARY OF INTRODUCED DATASETS

Dataset	Type	Nodes	Edges	Description
ca-HepTh	undirected	9877	25998	Collaboration network of High Energy Physics in Arxiv
GR-QC	undirected	5242	14496	Collaboration network of General Relativity in Arxiv
DBLP	undirected	$655K$	$2.0M$	Collaboration network from the DBLP Computer Science Bibliography Database
Facebook	undirected	4039	88234	Social circles network from Facebook
Flixster	directed	$1M$	$28M$	Network from Flixster

VI. APPLICATIONS

Influence maximization has far-ranging applications especially in viral marketing, rumour controls, and viral spreading control. For certain application, IM problems might be little different since the goal of various problems differs from each other. In viral marketing, the goal is to find the most influential users of products to achieve a maximum of the spread of the marketing. Usually, these applications are budgeted so, the targeted users should be highly cost-effective. As for rumour control, the goal is to block the spread of rumours which is opposite to viral marketing. The application on viral spreading is similar to rumour control whose goal is to control the viral spreading on the network(e.g. the scenario can be in real life or cyberspace).

VII. FUTURE DIRECTIONS

In future, the direction of IM problem studies might shift to the structure of the networks. The community of a network will gain more attention since networks are not only related to individuals but also a group of people. By studying the structure of networks, a large-scale network can be reconstructed to small-size networks containing fewer nodes and each node represents a community which is a small group of nodes connected together. Besides, learning methods on IM problems are recently worked. In future, more learning techniques might be adapted to this field and large-scale network analysis will become more intelligent.

VIII. CONCLUSION

This survey summarizes the models, techniques, datasets, and applications about the influence maximization problem. At first, the IM problem is formulated and simply analysed. The monotonicity and submodularity are also proven. Then models are summarised according to their behaviour. Following models, techniques are divided corresponding to the core idea inside. Datasets are also essential to know the details of the dataset will affect the design of models and algorithms. For the applications, influence maximization plays an important role in marketing since the influence propagation can contribute to the increment of profits. Of course, controlling influence spreading is also closely related to IM problem. At last, some future directions on this problem are discussed according to recent works.

ACKNOWLEDGMENT

The author would like to thank himself for setting before the computer for many days to type the whole survey. Besides, he also wants to thank his research tutor Ke Tang who encouraged him to bravely have a try on this survey. With friends going back home, the author felts more and more homesick but he chose to stay on campus for completing this survey. The determination of finishing this survey should be regarded as the greatest contributor.

REFERENCES

[1] P. Domingos, M. Richardson. Mining the Network Value of Customers. *Seventh International Conference on Knowledge Discovery and Data Mining*, 2001.

[2] M. Richardson, P. Domingos. Mining Knowledge-Sharing Sites for Viral Marketing. *Eighth Intl. Conf. on Knowledge Discovery and Data Mining*, 2002.

[3] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 03*, pages 137-146, New York, NY, USA, 2003.

[4] J. Goldenberg, B. Libai, E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters* 12:3(2001), 211-223.

[5] 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.

[6] M. Han, J. Li, Z. Cai, Q. Han, Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks. *IEEE SocialCom*, pp. 284-292, October 2016.

[7] E. Adar, L. Admic. Tracking information epidemics in blogspace. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI05)*, 2005.

[8] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology* 83(6):1420-1443, 1978.

[9] S. Liu, L. Ying, and S. Shakkottai, Influence maximization in social networks: An ising-model-based approach. In *Allerton*, 29 2010-oct. 1 2010, pp. 570-576.

[10] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[11] Q.B. Hu, G. Wang, and P.S. Yu. Transferring influence: Supervised learning for efficient influence maximization across networks. In *CollaborateCom*, 2014.

[12] Y.-C. Chen, W.-Y. Zhu, W.C. Peng, W.-C. Lee, and S.-Y. Lee, CIM: Community-based influence maximization in social networks. *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 25:1-25:31, 2014.

[13] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In *Proceedings of IEEE International Conference on Data Mining(ICDM)*, pages 918-923, 2012.

[14] X.R. He and D. Kempe. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 13-17, 2016, pages 885-894, 2016.

[15] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD* 2010.

[16] L.D. Fan et al., Least cost rumor blocking in social networks. In *Proceedings of IEEE 33rd International Conference on Distributed Computing Systems(ICDCS)*, Jul. 2013, pp. 540-549.

[17] Y. Li, D. Zhang, and K.-L. Tan, Real-time targeted influence maximization for online advertisements. In *Proceedings of the VLDB Endowment*, vol. 8, no. 10, 2015.

[18] 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.