Array 24 (2024) 100366

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Maximizing influence via link prediction in evolving networks

Kexin Zhang [a], Mo Li [b], Shuang Teng [c], Lingling Li [d,*], Yi Wang [a], Xuezhuan Zhao [e], Jinhong Di [a], Ji Zhang [f]

[a] School of Electronics and Information, Zhengzhou University of Aeronautics, Zhengzhou, 450046, China
[b] School of information, Liaoning University, Shenyang, 110036, China
[c] Henan Yuanfang Human Digital Technology Service Group Co., Ltd, Zhengzhou, 450000, China
[d] Zhengzhou University of Aeronautics, Zhengzhou, 450046, China
[e] School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou, 450046, China
[f] School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, 4350, Australia

ARTICLE INFO

Keywords:
Influence maximization
Future networks
Link prediction
Greedy algorithm
Graph query

ABSTRACT

Influence Maximization (IM), targeting the optimal selection of k seed nodes to maximize potential information dissemination in prospectively social networks, garners pivotal interest in diverse realms like viral marketing and political discourse dissemination. Despite receiving substantial scholarly attention, prevailing research predominantly addresses the IM problem within the confines of existing networks, thereby neglecting the dynamic evolutionary character of social networks. An inevitable requisite arises to explore the IM problem in social networks of future contexts, which is imperative for certain application scenarios. In this light, we introduce a novel problem, Influence Maximization in Future Networks (IMFN), aimed at resolving the IM problem within an anticipated future network framework. We establish that the IMFN problem is NP-hard and advocate a prospective solution framework, employing judiciously selected link prediction methods to forecast the future network, and subsequently applying a greedy algorithm to select the k most influential nodes. Moreover, we present SCOL (Sketch-based Cost-effective lazy forward selection algorithm Optimized with Labeling technique), a well-designed algorithm to accelerate the query of our IMFN problem. Extensive experimental results, rooted in five real-world datasets, are provided, affirming the efficacy and efficiency of the proffered solution and algorithms.

1. Introduction

In the era of advancing communication and electronic technologies, social networks – such as Twitter, Facebook, and Instagram – have witnessed exponential growth, seamlessly intertwining with the fabric of daily lives. This proliferation has facilitated the dissemination of information among network participants, engendering a plethora of applications spanning viral marketing, public opinion shaping, citizen safety education, and political opinion distribution, each wielding a substantial impact on everyday living. Within this milieu, scrutinizing the process of information spread in social networks garners paramount significance. Among the myriad research topics in this domain, Influence Maximization (IM) [1–4] emerges as one of the most fervent areas of study. The IM problem endeavors to select k nodes as seed nodes, with the objective of maximizing the spread of information (or influence) within social networks, all while adhering to a constrained budget.

The IM problems need to be solved under a specific influence diffusion model, the classic influence diffusion models including the Independent Cascade (IC) model [5,6], the Linear Threshold (LT) model [7], and the Weighted Cascade (WC) model [1], etc. Among them, the IC model is the most popular one due to its simplicity and applicability. Many methods have been proposed to solve IM problem based on these influence diffusion models. In [8], they categorized these methods into three groups: Simulation-based methods [9,10], Sketch-based methods [11–13], and Proxy-based methods [14–16]. However, to the best of our knowledge, most of the existing methods solve IM problems in current existing social networks, but in the real world, the topology of a social network evolves over time [17,18], and it is inevitable to solve influence maximization problem in future social networks sometimes. For instance, when formulating marketing plans for the upcoming year, predicting the network structure and selecting the most influential seed nodes in future networks become crucial for achieving better marketing effects. An illustration of a future network can be seen in Fig. 1, where

* Corresponding author.
E-mail address: lilingling@zua.edu.cn (L. Li).

https://doi.org/10.1016/j.array.2024.100366
Received 1 August 2024; Received in revised form 8 September 2024; Accepted 23 September 2024
Available online 28 September 2024
2590-0056/© 2024 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

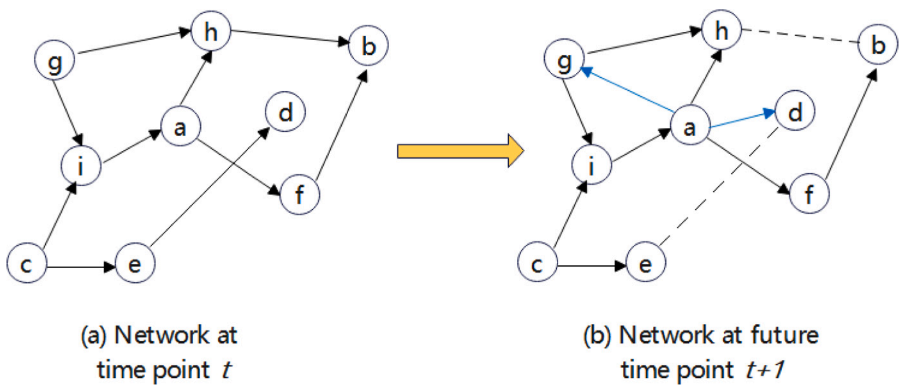

g
h
b
a
d
i
f
c
e
(a) Network at time point t
(b) Network at future time point t+1

Fig. 1. An example of future network. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 1(b) highlights the appearance of new links represented by blue lines and denotes disappearing links using dashed lines.

Based on the aforementioned discussion, we introduce the problem of Influence Maximization in Future Networks (IMFN). The objective of the IMFN problem is to identify the k most influential seed nodes within a future social network, presenting two primary challenges: (1) predicting the future period's social network topology, and (2) efficiently identifying the k most influential seed nodes within this future social network.

In this paper, we present a solution framework to address the dual challenges mentioned above. Our framework begins by utilizing a well-chosen link prediction technique proposed in [19] to predict the future social network; subsequently, a greedy algorithm is utilized to identify the k most influential seed nodes, moreover, to optimize the efficiency of greedy algorithm while ensuring quality, we also propose an improved algorithm called Sketch-based Cost-effective lazy forward selection algorithm Optimized with Labeling technique(SCOL) .

The contributions of this paper can be summarized as follows:

1. We define the IMFN problem aimed at identifying the k most influential seed nodes in a future social network. We have provided a precise definition, demonstrated that it is NP-hard under the Independent Cascade (IC) Model, and shown that the spread function $I(S, G_{t+1})$ is monotone and submodular.
2. We propose a predictive framework for solving the IMFN problem, which involves predicting the future social network using a well-chosen link prediction method, followed by solving the IMFN problem using a greedy algorithm.
3. To further enhance the efficiency of the aforementioned framework, we propose the well-designed SCOL algorithm optimized by a labeling technique. Notably, the SCOL algorithm provides theoretical guarantees for quality.
4. We provide extensive experimental results on five real-world datasets under two classic IC models, demonstrating the effectiveness and efficiency of the SCOL algorithm.

The rest of this paper is organized as follows: Section 2 introduces the notations used and provides an overview of related works. Section 3 defines the IMFN problem and offers a thorough analysis of the problem. The solution framework for IMFN problem is provided in Section 4. Section 5 describes the details of the proposed SCOL algorithm. Experimental results are presented in Section 6. Section 7 is the conclusion.

2. Preliminary

The notation frequently used in this article is summarized as follows.

In this paper, we define a directed graph G as $G = (V, E)$, where V is the vertex set and E is the associated edge set. Furthermore, G_{t+1} denotes the predicted graph at future time point $t + 1$, while $I(S, G)$ is the number of vertices(nodes) that are activated by S in graph G, $E(I(S, G))$ is the expected number of $I(S, G)$, OPT is the maximum expected spread of seed set whose size is k. θ is the number of generated subgraphs.

2.1. Independent Cascade (IC) model

IC model [5,6] is the most widely used influence propagation model due to its simplicity and applicability. Here we use a graph $G(V, E)$ to represent a social network and then describe the propagation process under the IC model. In graph G, nodes $V = (v_0, \dots v_i)$ represent the participants of the social network, and edges $E = (e_0, \dots e_j)$ indicate connections between these participants. At time point $t = 0$, a certain number of nodes are selected as the initial seed set S and are activated, while all other nodes remain inactive. In subsequent time points $t = i$, each newly activated node at time point $t = i - 1$ has one chance to activate its neighbors with a probability of $P(u, v)$. This process continues until no new nodes can be activated, then the propagation process is completed. The total number of nodes activated during this process is denoted as $I(S, G)$. From the discussion above, one can see that the propagation process under the IC model is very close to real-world propagation scenarios.

2.2. Link Prediction (LP)

Link prediction is a technique that aims to predict the existence of links between nodes in future networks by analyzing current network topology and node attributes, thus serving as valuable tools for predicting the future network structure. In [20], they classified the link prediction methods into three categories: similarity-based methods [21,22], maximum likelihood methods [20], probabilistic models based methods [23,24]. Among these, similarity-based methods are the prevailing approach, operating on the assumption that nodes with higher similarity are more likely to be connected in the future. Hence, the key to these methods lies in devising effective measures of node similarity. Nowadays, neural networks are widely utilized across various fields [25–27] due to their exceptional ability to learn nonlinear features. Researchers have increasingly explored different neural network architectures for link prediction tasks, in this paper, we employ the method proposed in [19] for future network prediction. This method first assigns labels to each node using the "label trick" technique and then feeds them into a graph neural network (GNN) which will learn the node representations, and finally address the link prediction problem effectively.

2.3. Influence Maximization(IM)

IM aims to find k nodes in a social network as the seed set, which will get the maximum information spread after information propagation under a certain influence diffusion model (e.g., IC model). Let

$I(S,G)$ be the number of vertices activated by S in graph snapshot G on the above influence propagation process under the IC model. The IM problem aims to find a size-k seed set S with the maximum expected spread $E(I(S,G))$. We define the IM problem as follows:

Definition 1 (*IM Problem [1]*)**.** Given a directed graph snapshot $G = (V,E)$, an integer k, the IM problem aims to find an optimal seed set S^* satisfying,

$$S^* = \underset{S \subseteq V, |S|=k}{\arg\max}\ E(I(S,G)) \tag{1}$$

Let OPT be the maximum expected spread of any size-k seed set, then we have $OPT = E(I(S^*,G))$. In Formula (1), $I(S,G)$ represents the information spread, which usually refers to the number of activated nodes after the information propagation process. Under the IC model, the propagation process is stochastic, and each node is activated according to a probability $P(u,v)$; thus we usually simulate the propagation process multiple times (e.g., 10,000 times), and then use the expected value $E[I(S,G)]$ to approximate the specific value of $I(S,G)$. Various algorithms have been proposed to solve the IM problem defined in Formula (1). Among these, the most classical and well-known approach is the Greedy algorithm [1], which selects a node with the maximum marginal spread gain in each round until the top k nodes are chosen. Although the Greedy algorithm is easy to interpret and has a theoretically guaranteed approximation ratio, its computational complexity is high, so various improved algorithms have been developed. One notable improved algorithm is the Cost-Effective Lazy Forward selection (CELF) algorithm proposed by [9]. CELF algorithm calculates the upper bound of each node's spread $I[\cdot]$ in the first iteration according to the submodularity, in subsequent iterations, many nodes with small spread $I[\cdot]$ are pruned, thus significantly reducing computational complexity. Compared with the Greedy algorithm, CELF achieves a performance improvement of approximately 700 times. Later, C. Zhou et al. [10] proposed the UBLF method, which gave a tighter upper bound for each node's spread $I[\cdot]$ based on the propagation probability matrix, further reducing the computational complexity.

Different from the simulation-based methods mentioned above, S. Cheng et al. [11] presented the StaticGreedy method, which replaces Monte Carlo simulations with a certain number of sketch subgraphs, resulting in computational efficiency improvement, the improved StaticCELF method was also proposed in the same work to further improve the performance of StaticGreedy. Later, the representative reverse reachable sketch-based algorithms [12,13,28] were proposed and gained widespread attention. These reverse reachable sketch-based algorithms operate under the assumption that the more influential a node is, the more nodes can be reached from it, based on this assumption, C. Borgs et al. [12] first defined the Reverse Reachable Set (RRS). After obtaining a sufficient number of random RRS, the k nodes that cover the most RRS are selected as the seed set. Later, Y. Tang et al. [13,28] presented TIM/IMM, which provided a lower boundary for the number of RRS needed to guarantee the quality. It should be noted that although the reverse reachable sketch-based algorithms are faster, they require large memory spaces to store the RRS. All the methods mentioned above have a theoretical guarantee for quality. There are also some proxy-based methods that are highly efficient. DegreeDiscount in [14] adopted degree discount heuristics to handle the influence overlaps, which improve the performance of initial degree-based methods. NCVoteRank in [16] employed the coreness based VoteRank method, which takes into account the voting abilities of different nodes to select the most influential nodes. FIP in [29] extracted the candidate nodes based on community structure, which decreased the search space for selecting final seed nodes. Although these methods are efficient, they lack quality guarantees and their performance may fluctuate in different scenarios. Meanwhile, there are some works aimed at solving the dynamic influence maximization (DIM) problem. H. Zhuang et al. [30] developed an Maximum Gap Probing (MaxG) algorithm to obtain a partially observed dynamic network by probing a subset of nodes in a social network, then solved the DIM problem on the partially observed dynamic networks. Y. Wang et al. [31] proposed the Stream Influence Maximization (SIM) problem, which aimed to maintain the most influential k seeds over dynamic networks by adopting a sliding window model. B. Peng et al. [32] studied the DIM problem under incremental model and fully dynamic model, then provided the boundaries of running time and approximation rate for both models. However, these works primarily address the IM problem on known or partially known networks.

To the best of our knowledge, most of the methods mentioned earlier are applied to solve the IM problem in current existing network. However, in this paper, we aim to address the Influence Maximization problem in Future Networks(IMFN), the definition and detailed discussion of the IMFN problem will be given in Section 3.

3. Problem definition

In this section, we provide the definition of the IMFN problem and analyze it in detail.

Definition 2 (*IMFN Problem*)**.** Give a social network $G_i = (V_i, E_i)$ whose topology evolves over time. If we can predict its' network topology $G_{t+1} = (V_{t+1}, E_{t+1})$ in the future time point $t+1$, the IMFN problem aims to choose k nodes in G_{t+1} as the seed node set S^* which will get the maximum expected information spread $E(I(S,G_{t+1}))$ under a specific influence diffusion model (e.g. IC model). The IMFN problem can be formulated using the following equation:

$$S^* = \underset{S \subseteq V_{t+1}, |S|=k}{\arg\max}\ E(I(S,G_{t+1})) \tag{2}$$

Theorem 1 (*NP-hard*)**.** *The IMFN problem defined in Definition 2 is NP-hard under the IC model.*

Proof. Since the Maximum Cover (MC) problem [33] is NP-hard, if it is reducible to IMFN problem in polynomial time, then the IMFN problem is NP-hard; thus, we need to find a solution to reduce the MC problem to IMFN problem in polynomial time to prove the hardness of IMFN problem under IC model. During the dissemination process under IC model,each node v is activated by its newly activated neighbors with a probability of $P(u,v)$, which is a random event. This random event can be determined by flipping a coin. According to the analysis in [1], we can perform the coin-flip operation at the very beginning of the dissemination process, which is equivalent to doing it during the process. Thus, for the future network $G_{t+1} = (V_{t+1}, E_{t+1})$, we perform the coin-flip operation at the very beginning for each edge $e_j \in E_{t+1}$ to determine whether the edge is preserved. Subsequently, all the nodes $v_i \in V_{t+1}$ and the preserved edges form a directed graph G'. The spread of a node v_i can be measured as the number of its children in G'. In this stage, we define a set $S = S_1, S_2, S_3 ... S_{|V_{t+1}|}$, where each set S_i corresponds to a node v_i in V_{t+1}, and the elements of S_i are the children of node v_i in G'. We also define a set U whose elements are all the nodes in V_{t+1}. The IMFN problem, which aims to select k nodes from V_{t+1} as the seed node set S^* that maximizes the spread, is equivalent to the MC problem which choosing k set in $S = S_1, S_2, S_3 ... S_{|V_{t+1}|}$ to cover as many elements in U as possible, thus the MC problem is reduced to the IMFN problem. Since the above reduction can be done in polynomial time, we conclude that the IMFN problem is NP-hard under IC model. Theorem 1 is proved. □

Theorem 2 (*Monotonicity and Submodularity*)**.** *The spread function $I(S,G_{t+1})$ in IMFN problem is monotone and submodular, this implies that if a seed set $S_a \subset S_b$, then $I(S_a,G_{t+1}) \leq I(S_b,G_{t+1})$, and for a node $v_j \in V_{t+1} \setminus S_a$, $I(S_a \cup v_j, G_{t+1}) - I(S_a,G_{t+1}) \geq I(S_b \cup v_j, G_{t+1}) - I(S_b,G_{t+1})$.*

Proof. At the very beginning of the dissemination process under IC model in future network $G_{t+1} = (V_{t+1}, E_{t+1})$, we perform the same coin-flip operation as that in the proof of Theorem 1, then we get a directed graph G'. The spread $I(S, G')$ of a seed set S can be denoted as the number of its children in G'. For a seed set $S_a \subset S_b$, it is evident that $I(S_a, G') \leq I(S_b, G')$. Thus, the spread function $I(S, G')$ is monotone. Furthermore, since the non-negative linear combination of monotone functions is also monotone, the spread function $I(S, G_{t+1})$ in IMFN problem is also monotone.

Inspired by the work in [1], we adopt the same method to prove that the spread function $I(S, G_{t+1})$ in IMFN problem is submodular. Referring to the same graph G' mentioned earlier, we define the set S^{child} as the set of children nodes of set S, and V_j^{child} as the set of children nodes of node v_j. For a node $v_j \in V_{t+1} \setminus S_a$ and set $S_a \subset S_b$, the marginal gain of node v_j with respect to set S_a is denoted as $I(S_a \cup v_j, G') - I(S_a, G')$, whose value is equal to $|V_j^{child}| - |S_a^{child} \cap V_j^{child}|$. Similarly, the marginal gain of node v_j with respect to set S_b is denoted as $I(S_b \cup v_j, G') - I(S_b, G')$, whose value is equal to $|V_j^{child}| - |S_b^{child} \cap V_j^{child}|$, it is obvious that $|S_b^{child} \cap V_j^{child}| \geq |S_a^{child} \cap V_j^{child}|$, thus $I(S_a \cup v_i, G') - I(S_a, G') \geq I(S_a \cup v_j, G') - I(S_a, G')$. Consequently, the spread function $I(S, G')$ is submodular. Moreover, since the non-negative linear combination of submodular functions is also submodular, the spread function $I(S, G_{t+1})$ in IMFN problem is submodular. Hence, Theorem 2 is proven. □

According to Theorem 1, the IMFN problem is NP-hard,thus one cannot solve IMFN problem in polynomial time. Fortunately based on Theorem 2, the spread function $I(S, G_{t+1})$ in IMFN problem is monotone and submodular,so we can take advantage of these properties of $I(S, G_{t+1})$ and then give the approximation of optimal solution by using the Greedy algorithm.

4. The basic framework for IMFN

Based on the definition and analysis in Section 3, we now introduce the basic solution framework for IMFN problem. The framework consists of two steps: (1) utilizing the well-chosen link prediction method to predict the future network topology $G_{t+1} = (V_{t+1}, E_{t+1})$ in the future time point $t + 1$; (2) employing the Greedy algorithm to obtain the seed set S, which is the approximation of the optimal seed set S^*. The detailed process is described in Algorithm 1.

In Algorithm 1, we first predict the future network G_{t+1} by using the link prediction method proposed in [19] (line 1). Next, the greedy algorithm is utilized to select the node $v_j \in V_{t+1} \setminus S$ with the maximum marginal spread gain in each iteration(line 4–7). Here V_{t+1} represents the nodes in the graph G_{t+1}, while S denotes the selected seed node set which is initially empty. The marginal gain is denoted by $M(v_j)$(line 6). Subsequently, the selected node v_j is added to the seed set S (line 8). This process continues until k seed nodes are chosen. The greedy algorithm described in Algorithm 1 guarantees an approximation of the optimal solution with a factor of $(1 - 1/e)$ which can be expressed by Inequality (3).

$$I(S, G_{t+1}) \geq (1 - 1/e) I(S^*, G_{t+1}) \tag{3}$$

In Inequality (3), S^* represents the optimal seed set, and S denotes the seed set selected using Algorithm 1. Although the greedy algorithm presented in Algorithm 1 provides a theoretical guarantee for solving IMFN problem, it suffers from a significant drawback: its computational complexity is excessively high, severely limiting its applicability, even in medium-sized networks.

Next, we will analyze the computational complexity of Algorithm 1. The computation of this algorithm comes from two parts: (1) predicting the future network using link prediction, and (2) the greedy algorithm employed for selecting seed sets. Since predicting the future network does not require excessive time, we will focus on Part 2. For the future network at time point $t+1$, denoted as $G_{t+1} = (V_{t+1}, E_{t+1})$, according to

Algorithm 1: IFMN Algorithm

```
Input: An evolving network $G_i$, an integer number $k$(the size of
       seed set $S$), $R$(the number of Monte Carlo simulations).
Output: The selected seed set $S$ for future network $G_{t+1}$.
1  Predict the future network $G_{t+1} = (V_{t+1}, E_{t+1})$ in time point $t+1$
     by link prediction method in [19];
2  $S \leftarrow \emptyset$;
3  for $i = 1 : k$ do
4      for $j = 1 : |V_{t+1}|$ do
5          Calculating the marginal gain of node
             $v_j \in V_{t+1} \setminus S$,which is denoted by $M(v_j)$;
6          $M(v_j) \leftarrow spreadmc(S \cup v_j, G_{t+1}) - spreadmc(S, G_{t+1})$;
7      Select node with the maximum marginal gain $M(\cdot)$ as the
         seed node $v^*$;
8      $S \leftarrow S \cup v^*$
9  return $S$
10 Function spreadmc($S$,$G_{t+1}$)
11     New active set $NS \leftarrow S$;
12     Actived Set $AS \leftarrow S$;
13     Influence=0;
14     for $i = 1 : R$ do
15         while $NS \neq \emptyset$ do
16             Each node $v_i$ in $NS$ have a chance of $P(u, v)$ to
                 activate its neighbors;
17             Add the activated nodes into $AS$ and $NS$;
18             Delete $v_i$ from $NS$;
19         Influence ← Influence + $|AS|$;
20     return Influence/R ;
```

Algorithm 1, we need to repeat k iterations(line 5) to select k nodes as the seed set S. Within each iteration, there are at most $|V_{t+1}|$ nodes that need to calculate the marginal spread gain $M(\cdot)$. Additionally, due to the stochastic nature of the propagation process, when calculating the marginal spread gain $M(\cdot)$ of each node, we need to perform R Monte Carlo simulations to obtain the expectation of the marginal spread gain $M(\cdot)$. In each Monte Carlo simulation, there are at most $|E_{t+1}|$ edges that need to be simulated. Consequently, the overall complexity of the greedy algorithm in Algorithm 1 is $\mathcal{O}(k \cdot |V_{t+1}| \cdot R \cdot |E_{t+1}|)$. The value of R should be large enough to ensure the submodularity property of the spread function $I(\cdot)$, typically set to 10 000 [1]. As we can see, the complexity of Algorithm 1 is too high to be practically applicable. Hence, we proposed the Sketch based CELF Optimized with Labeling technique (SCOL) algorithm in Section 4 to improve the efficiency of Algorithm 1.

5. Sketch based CELF optimized with labeling technique

Based on the complexity analysis in Section 4, there are three approaches to reduce the computational complexity of Algorithm 1. The first is to reduce R which is the number of Monte Carlo simulations; the second is to reduce the number of candidate nodes that need to calculate the marginal spread gain, corresponding to $|V_t|$; and the third is to reduce the computational complexity of calculating the marginal spread gain for each candidate node. Taking all of these approaches into consideration, we propose the Sketch based CELF Optimized with Labeling technique(SCOL) algorithm in this section. The SCOL algorithm combines sketch technique, CELF method and is optimized with a labeling technique to enhance the efficiency of Algorithm 1.

5.1. Sketch technique

The Monte Carlo simulation process under IC model can be described as follows: In a social network $G = (V, E)$ with the seed set S, for each newly activated node v_j in V with a probability of $P(u,v)$ to activate its' inactive neighbor, we generate a random number between $(0,1)$, if the number is not greater than $P(u,v)$, then the neighbor becomes active; otherwise, it stays inactive. Continue this process until there are no new nodes that can be activated. At the end of the propagation, the total number of active nodes represents the spread $I(S,G)$. Due to the stochastic nature of the propagation process, we need to perform the Monte Carlo simulation R times (typically 10 000 times) to obtain the average spread $E(I(S,G))$ as the final result.

The reason why so many Monte Carlo simulations are required to be performed is that we need to ensure the submodularity of the spread function $I(S,G)$ to guarantee the approximation rate to the optimal solution. S. Cheng et al. [11] pointed out that we can employ the sketch technique to maintain the submodularity instead of conducting numerous Monte Carlo simulations. In the sketch technique proposed by [11], for a social network $G=(V,E)$, a subgraph of G is constructed by deleting each edge e_j in E with a probability of $1-P(u,v)$, denoted as a sketch G^i, then repeating the process to construct θ sketches $G^0,G^1...G^\theta$. For a given seed set S, its' spread can be calculated by averaging the numbers of nodes reached by S in each sketch. Later, Y. Li et al. [8] pointed out that θ should satisfy In Eq. (4) to ensure a $(1-1/e-\epsilon)$ approximation rate of the optimal solution with a probability of $(1-1/|V|)$. Here, ϵ is the sampling error.

$$\theta \geq (8+2\epsilon)\cdot|V|\cdot\frac{\log|V|+\log\binom{|V|}{k}+log2}{\epsilon^2} \tag{4}$$

5.2. CELF

The CELF algorithm [9] leverages the submodularity of the spread function $I(\cdot)$, which implies the marginal spread gain of a node v in each round is not greater than its marginal spread gain in the previous round, so when selecting the node with the maximum marginal spread gain in each round, the calculation process of many nodes with smaller marginal spread gain in the previous round is pruned, thus significantly reducing the computational workload. In this section, we will use the CELF method to reduce the number of candidate nodes that need to calculate the marginal spread gain, which will accelerate the speed of the process of selecting the most influential nodes.

5.3. Labeling technique

By observing the process of calculating marginal spread gain in the sketch-based method, we found that when calculating the marginal spread gain of a node v corresponding to seed sets S, we need to calculate the spread $I(S)$ of set S, and then calculate the spread $I(S\cup v)$ of $S\cup v$. The marginal spread gain of node v is denoted by Formula (5)

$$M(v)=I(S\cup v)-I(S) \tag{5}$$

Upon careful examination of the calculation process, we found that it involved additional calculations for the spread of seed set S, which greatly increased the computational cost of calculating the marginal spread gain of node v. Based on the observation above, if we can reduce the additional computation of spread of S, we can reduce the computation in the process of calculating marginal spread gain. Inspired by the work in [34], we use a labeling technique to reduce the additional computation of spread of S. Specially, in each sketch, we use a flag to mark whether a node is the child of set S; if yes, $flag=0$, otherwise, $flag=1$. When we calculate the marginal spread gain of a node v, we adopt a Breadth First Search(BFS) strategy to find the children $v_{children}$ of v, during which the node with flag equal to 0 will not be involved in $v_{children}$, then we get the margin spread gain of v, which is the size of $v_{children}$.

Combining the techniques mentioned in Sections 5.1, 5.2, and 5.3, we proposed the Sketch based CELF Optimized with Labeling technique(SCOL) algorithm in this section to enhance the efficiency of Algorithm 1. The pseudo-code of SCOL is shown in Algorithm 2. Due to space limitations, the three sub-functions used in Algorithm 2 are shown in Algorithm 3.

Algorithm 2: SCOL Algorithm

```
Input: An evolving network $G$, an integer number $k$(the size of seed set $S$) .
Output: The seed set $S$ for future network $G_{t+1}$.
1  Predict the future network $G_{t+1}=(V_{t+1},E_{t+1})$ in time point $t+1$ by link prediction method in [19]
2  $S\leftarrow\emptyset$ ;
3  Repeatedly Construct $\theta$ sketches $\{G^0_{t+1},G^1_{t+1}...G^\theta_{t+1}\}$ by deleting each edge $e_j$ in $E_{t+1}$ with a probability of $(1-P(u,v))$ ;
4  for $i\leftarrow 1:\theta$ do
5      for $node\in G^i_{t+1}$ do
6          $Flag[node][i]\leftarrow 1$ ;
7  for $i\leftarrow 1:|V_{t+1}|$ do
8      $M(v_i)\leftarrow spread(v_i,\theta,\{G^0_{t+1},G^1_{t+1}...G^\theta_{t+1}\})$ ;
9  $v^*\leftarrow \arg\max_{v_i\subseteq V_{t+1}} M(v_i)$ ;
10 $S\leftarrow S\cup v^*$ ;
11 $Label(v^*,\theta,Flag,\{G^0_{t+1},G^1_{t+1}...G^\theta_{t+1}\})$ ;
12 for $i\leftarrow 2:k$ do
13     while 1 do
14         Select the node $v_m\in V_{t+1}\setminus S$ with the maximum marginal spread gain $M(\cdot)$ ;
15         $M(v_m)\leftarrow MGspread(v_m,\theta,Flag,\{G^0_{t+1},G^1_{t+1}...G^\theta_{t+1}\})$ ;
16         if $M(v_m)$ is the largest one among all the node $v\in V_{t+1}\setminus S$ then
17             Jump out of current while loop
18     $S\leftarrow S\cup v_m$ ;
19     $Label(v_m,\theta,Flag,\{G^0_{t+1},G^1_{t+1}...G^\theta_{t+1}\})$ ;
20 return S
```

In Algorithm 2, the future network G_{t+1} is predicted using link prediction method in [19](Line 1), then we construct θ sketches (Line 3) based on the network G_{t+1}. According to [8], the value of θ should satisfy the In Eq. (4) to ensure a $(1-1/e-\epsilon)$ approximation rate of the optimal solution. The flag of each node in each sketch is initialized to be 1 (Lines 4–6). We calculate the marginal spread gain of each node (Lines 7–8) as its' upper boundary and select the node with the maximum marginal spread gain as the first seed node to be added to the seed set S (Lines 9–10). Next, we repeat the following operations in each round: select the node $v_m\in V_{t+1}\setminus S$ with the maximum marginal spread gain $M(\cdot)$ and update its marginal spread gain which is denoted as $M(v_m)$ by using Function $MGspread(\cdot)$ shown in 3 (Lines 14–15), if $M(v_m)$ is the largest one among all the nodes $v\in V_{t+1}\setminus S$, jump out of the current *while* loop, and add the node v_m to the seed set S (Line 18). Otherwise, continue with the current *while* loop. Each time we select a seed node, the flag $Flag[\cdot][\cdot]$ of all nodes in each sketch should be updated (Line 11 and Line 19) by using the Function $Label(\cdot)$ in Algorithm 3. Finally, after $k-1$ rounds, k seed nodes are selected as the seed set S. As observed, the proposed SCOL method utilizes the sketch technique to reduce the number of Monte Carlo simulations and adopts the CELF method to minimize the number of candidate nodes required for calculating the marginal spread gain, also employs the labeling technique to accelerate the speed of calculating the marginal spread gain of each candidate node. Consequently, the SCOL algorithm significantly decreases the computational complexity, which is demonstrated by the experimental results in Section 6.

Algorithm 3: Three sub-functions used in Algorithm 2

```
Function spread($S, \theta, \{G^0_{t+1}, G^1_{t+1} ... G^\theta_{t+1}\}$)
    influence ← 0 ;
    for $i \leftarrow 1 : \theta$ do
        children ← The number of nodes reached by S in $G^i_{t+1}$;
        influence ← influence + children ;
    return influence/$\theta$ ;
Function Label($v, \theta, Flag, \{G^0_{t+1}, G^1_{t+1} ... G^\theta_{t+1}\}$)
    for $i \leftarrow 1 : \theta$ do
        $Queue \leftarrow v$;
        while Queue not empty do
            $nodecur \leftarrow Queue.pop$ ;
            for $Neighbor \in$ the direct children of nodecur do
                if Neighbor is not visited then
                    $Flag[Neighbor][i] \leftarrow 0$;
                    $Queue.append(Neighbor)$;
Function MGspread($v, \theta, Flag, \{G^0_{t+1}, G^1_{t+1} ... G^\theta_{t+1}\}$)
    influence ← 0 ;
    for $i \leftarrow 1 : \theta$ do
        if $v \in G^i_{t+1}$ then
            if $Flag[v][i] \neq 0$ then
                influence ← influence+1 ;
                while Queue not empty do
                    $nodecur \leftarrow Queue.pop$ ;
                    for $Neighbor \in$ the children of nodecur do
                        if $Flag[Neighbor][i] \neq 0$ then
                            if Neighbor is not visited then
                                influence ← influence+1 ;
                                $Queue.append(Neighbor)$;
    return influence/$\theta$ ;
```

6. Experimental result

In this section, we present the experimental results for IFMN problem based on the widely used IC model. Four state-of-the-art algorithms: StaticCELF (denoted as SC) [11], IMM [13], DegreeDiscount (denoted as DD) [14], and NCVoteRank (denoted as NCVR) [16] were chosen as the baselines for comparison. Five real-world datasets[1]: *EmailEuCore*, *AskUbuntu*, *Superuser*, *Wikitalk* and *StackOverflow* were used to conduct the experiments. The experiment results were presented from two perspectives: (1) effectiveness, measured by the total spread (number of nodes activated) achieved by the selected seed set S, (2) efficiency, denoted by the running time of each method. The experiments were conducted on an HP workstation equipped with an i7-12700 processor, 32 GB of memory, and an RTX3060 graphics card. The software environment used was Python 3 based on the Windows 10 operating system.

6.1. Datasets and parameters setting

The selected five real-world datasets are typical temporary networks whose properties are shown in Table 1. In particular, the datasets *Wikitalk* and *StackOverflow* are too large to be simulated on our experimental platform. Therefore, for these two datasets, we extracted the data of 1260 days and 400 days for our experiments, respectively. All selected datasets were divided into 20 snapshots, with each snapshot corresponding to a specific time point $t = i$. We then utilized the link prediction method proposed in [19] to predict the future network $G_{t+1} = (V_{t+1}, E_{t+1})$. To evaluate the effectiveness and efficiency of the five algorithms, we set the size of the selected seed set S to be 10, 20, 30, 40, and 50, respectively. For the StatciCELF and SCOL algorithms, θ was set to 200. For the IC model, two classic models: trivalency model [35] and constant model [1,35] were adopted. In trivalency model, the probability of each edge was randomly selected from 0.001, 0.01, and 0.1, while in constant model, the probability of each edge was empirically set to be 0.03. The total spread of S selected by each algorithm was obtained by 10,000 Monte Carlo simulations.

Table 1
Dataset characteristic.

Name	Type	Nodes	Temporal edges	$Degree_{avg}$	Time span
EmailEuCore	Directed, temporal	986	332 334	25.28	803
AskUbuntu	Directed, temporal	79 155	327 513	2.01	2047
Superuser	Directed, temporal	94 548	479 067	2.52	2769
Wikitalk	Directed, temporal	28 491	293 329	3.6	1260
StackOverflow	Directed, temporal	51 989	706 676	12.10	400

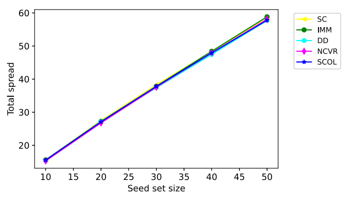
6.2. Effectiveness evaluation

In this section, we evaluate the effectiveness of five approaches in terms of total spread (the number of activated nodes), using different datasets and varying k under trivalency model and constant model, respectively. Fig. 2 illustrate the average total spread achieved by five algorithms, with k varying from 10 to 50 across five distinct datasets under trivalency model, and Fig. 3 is the spread achieved under constant model. The vertical axis represents the total spread, and the horizontal axis represents the size of the selected seed sets. From Fig. 2 we can see that the proposed SCOL algorithm always has almost the same total spread as the state-of-the-art algorithms IMM and StaticCELF which have theoretical quality guarantees, they have the largest total spread among all the algorithms used in our experiment. Then is the NCVoteRank algorithm, which has almost the same total spread as SCOL on *EmailEuCore* and *AskUbuntu* datasets, while its' total spread is slightly worse than SCOL on datasets *Superuser* and *StackOverflow*, and the decrease in total spread for NCvoteRank is noticeable on dataset *Wikitalk*. For the DegreeDiscount algorithm, it has almost the same total spread as SCOL on datasets *EmailEuCore*, *AskUbuntu* and *Superuser*. However, the total spread decreases rapidly on datasets *StackOverflow* and *Wikitalk*. From Fig. 2, we can see that there is a similar trend under the constant model. All of these demonstrate the effectiveness of the proposed SCOL. This can be attributed to the fact that SCOL combines the sketch technique, CELF method and is optimized with a labeling technique, which provide theoretical guarantees for high quality.

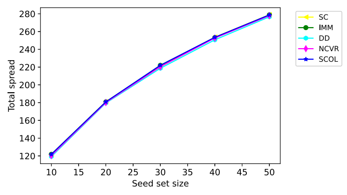
6.3. Efficiency evaluation

In this section, we examine the efficiency of five approaches in terms of running time, utilizing different datasets while varying k under trivalency model and constant model, respectively. Fig. 4 shows the average running time of five algorithms by varying k from 10 to 50 in five different datasets under trivalency model, and Fig. 5 shows the average running time under constant model. The vertical axis represents the average running time, and the horizontal axis represents the size of the selected seed sets. From Fig. 4, we can see that the DegreeDiscount algorithm consumes the least time, followed by NCVoteRank. However, as we discussed in Section 6.2, these two algorithms do not have a theoretical quality guarantee and exhibit significant fluctuations in performance on different datasets. Among the three algorithms StaticCELF, IMM, and SCOL which have theoretical quality guarantees,

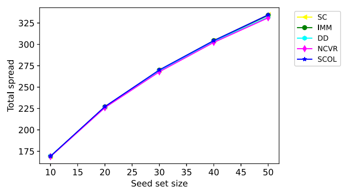
[1] http://snap.stanford.edu/data/index.html.



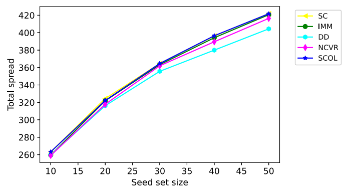
(a) EmailEuCore



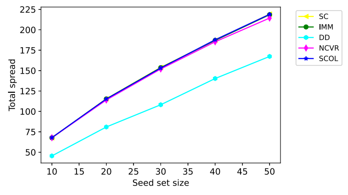
(b) AskUbuntu



(c) Superuser

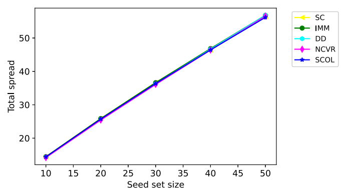


(d) Wikitalk

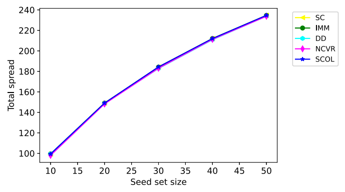


(e) StackOverflow

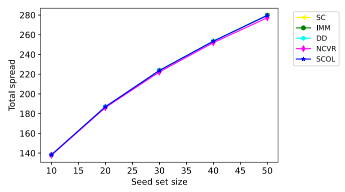
Fig. 2. Total spread on different datasets with various K under trivalency model.



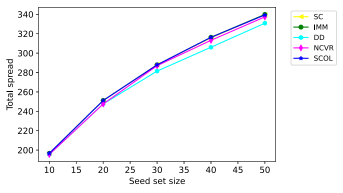
(a) EmailEuCore



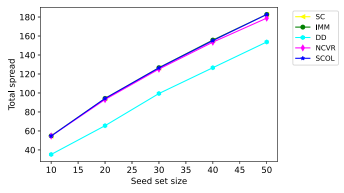
(b) AskUbuntu



(c) Superuser

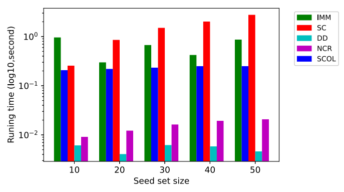


(d) Wikitalk

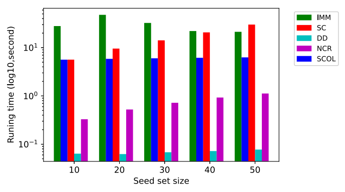


(e) StackOverflow

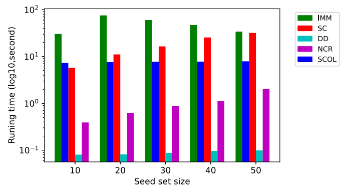
Fig. 3. Total spread on different datasets with virious K under constant model.



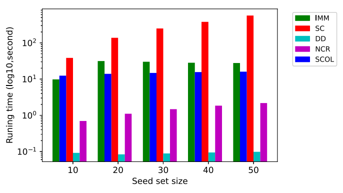
(a) EmailEuCore



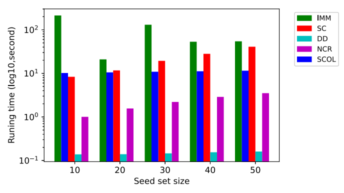
(b) AskUbuntu



(c) Superuser



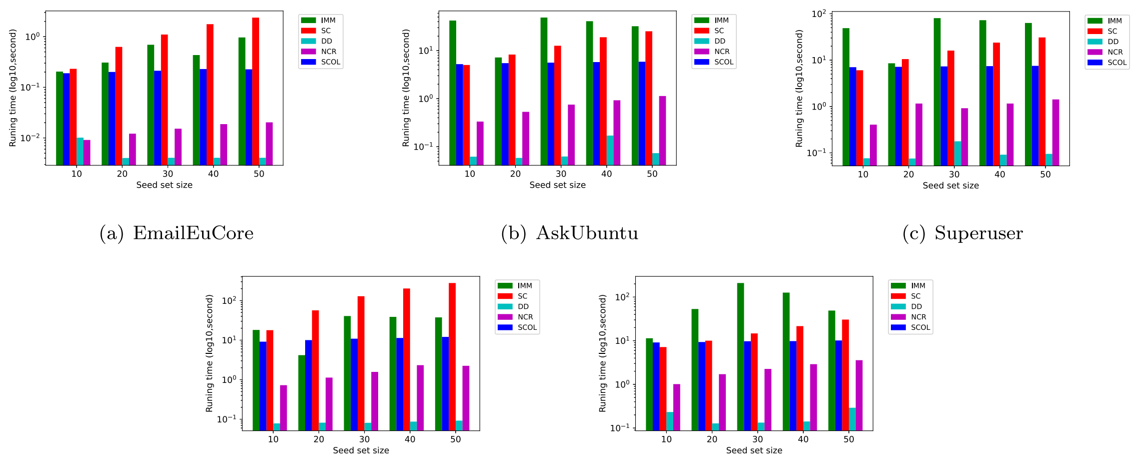
(d) Wikitalk



(e) StackOverflow

Fig. 4. Time cost of algorithms with varying k under trivalency model.

SCOL consumes the least time in almost all the cases, except for $N = 10$, where it consumes slightly more time than IMM on datasets *Wikitalk* and StaticCELF on datasets *Superuser*, *StackOverflow*. This is because when N is relatively small, the advantages of the labeling



(a) EmailEuCore (b) AskUbuntu (c) Superuser (d) Wikitalk (e) StackOverflow

Fig. 5. Time cost of algorithms with varying k under constant model.

technique used by SCOL are not fully reflected. Particularly, SCOL is at most 10.06 times, 3.77times, 3.04 times, 34.46 times, and 2.55 times faster than StaticCELF on datasets *EmailEuCore*, *AskUbuntu*, *Superuser*, *Wikitalk* and *StackOverflow*, respectively, and is at most 3.66 times, 7.16 times, 8.98 times, 1.26 times, and 19.88 times faster than IMM on datasets *EmailEuCore*, *AskUbuntu*, *Superuser*, *Wikitalk* and *StackOverflow*, respectively. We can find a similar trend in Fig. 5, which is under the constant model. Particularly, SCOL is at most 9.5 times, 3.3 times, 3.1 times, 22.03 times, and 2.01 times faster than StaticCELF on datasets *EmailEuCore*, *AskUbuntu*, *Superuser*, *Wikitalk* and *StackOverflow*, respectively, and is at most 3.26 times, 7.67 times, 10.01 times, 2.75 times, and 20.6 times faster than IMM on datasets *EmailEuCore*, *AskUbuntu*, *Superuser*, *Wikitalk* and *StackOverflow*, respectively. Furthermore, from Figs. 4 and 5, we can see that as N increases, the time consumed by SCOL does not significantly increase. This can be attributed to the fact that as N increases, the CELF method and labeling technique adopted by SOCL can greatly reduce the number of candidate nodes and the computational cost of calculating the marginal spread gain for each candidate node, respectively. All of these illustrate the superiority of the proposed SCOL in terms of efficiency.

7. Conclusion

In this paper, we studied the problem of Influence Maximization in Future Networks (IMFN). The IMFN problem aims to solve the influence maximization problem in future networks, which is critical for some important applications such as viral marketing, political opinion dissemination, etc. We proposed a basic solution framework for the IMFN problem, which utilizes a well-chosen link prediction method to predict the future network and then applies the greedy algorithm to select the k most influential nodes as the seed set S. Furthermore, we introduced the SCOL algorithm to improve the efficiency of the solution. Finally, experimental results based on five real-world datasets were provided to illustrate the effectiveness and efficiency of the proposed algorithm SCOL. In the future, our focus will be on researching more accurate link prediction methods and developing even more efficient solutions to tackle the IMFN problem, especially for super large-scale networks.

CRediT authorship contribution statement

Kexin Zhang: Writing – original draft, Validation, Methodology, Conceptualization. **Mo Li:** Visualization, Resources, Formal analysis, Data curation. **Shuang Teng:** Software, Data curation. **Lingling Li:** Writing – review & editing, Supervision, Funding acquisition. **Yi Wang:** Writing – review & editing, Data curation. **Xuezhuan Zhao:** Software, Resources. **Jinhong Di:** Software, Investigation. **Ji Zhang:** Writing – review & editing, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by Henan Province key Research and Development Project, China (231111212000), Henan Center for Outstanding Overseas Scientists, China (GZS2022011), Research Programs of Henan Science and Technology Department, China (2321022 10054, 242102211013), Program for Science and Technology Innovation Talents in Universities of Henan Province, China under Grand (22HASTIT020), Chongqing Natural Science Foundation, China (2023 NSCQ-MSX4618), Henan International Joint Laboratory of Aerospace Intelligent Technology and Systems, China.

References

[1] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining. 2003, p. 137–46.
[2] Ou J, Buskens V, Van De Rijt A, Panja D. Influence maximization under limited network information: seeding high-degree neighbors. J Phys: Complexity 2022;3(4):045004.
[3] Cai T, Lei Q, Sheng QZ, Cui N, Yang S, Yang J, Zhang WE, Mahmood A. Reconnecting the estranged relationships: Optimizing the influence propagation in evolving networks. IEEE Trans Knowl Data Eng 2023.
[4] Tsaras D, Trimponias G, Ntaflos L, Papadias D. Collective influence maximization for multiple competing products with an awareness-to-influence model. Proc VLDB Endow 2021;14(7):1124–36.
[5] Goldenberg J, Libai B, Muller E. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Mark Lett 2001;12:211–23.
[6] Du N, Song L, Gomez Rodriguez M, Zha H. Scalable influence estimation in continuous-time diffusion networks. Adv Neural Inf Process Syst 2013;26.
[7] Granovetter M. Threshold models of collective behavior. Am J Sociol 1978;83(6):1420–43.
[8] Li Y, Fan J, Wang Y, Tan K-L. Influence maximization on social graphs: A survey. IEEE Trans Knowl Data Eng 2018;30(10):1852–72.
[9] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. 2007, p. 420–9.

[10] Zhou C, Zhang P, Zang W, Guo L. On the upper bounds of spread for greedy algorithms in social network influence maximization. IEEE Trans Knowl Data Eng 2015;27(10):2770–83.
[11] Cheng S, Shen H, Huang J, Zhang G, Cheng X. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of the 22nd ACM international conference on information & knowledge management. 2013, p. 509–18.
[12] Borgs C, Brautbar M, Chayes J, Lucier B. Maximizing social influence in nearly optimal time. In: Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms. SIAM; 2014, p. 946–57.
[13] Tang Y, Shi Y, Xiao X. Influence maximization in near-linear time: A martingale approach. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data. 2015, p. 1539–54.
[14] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. 2009, p. 199–208.
[15] Galhotra S, Arora A, Roy S. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In: Proceedings of the 2016 international conference on management of data. 2016, p. 743–58.
[16] Kumar S, Panda B. Identifying influential nodes in social networks: Neighborhood coreness based voting approach. Phys A 2020;553:124215.
[17] Chen X, Song G, He X, Xie K. On influential nodes tracking in dynamic social networks. In: Proceedings of the 2015 SIAM international conference on data mining. SIAM; 2015, p. 613–21.
[18] Leskovec J, Backstrom L, Kumar R, Tomkins A. Microscopic evolution of social networks. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. 2008, p. 462–70.
[19] Zhang M, Li P, Xia Y, Wang K, Jin L. Labeling trick: A theory of using graph neural networks for multi-node representation learning. Adv Neural Inf Process Syst 2021;34:9061–73.
[20] Lü L, Zhou T. Link prediction in complex networks: A survey. Phys A 2011;390(6):1150–70.
[21] Liu W, Lü L. Link prediction based on local random walk. Europhys Lett 2010;89(5):58007.
[22] Zhou T, Lee Y-L, Wang G. Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. Phys A 2021;564:125532.
[23] Taskar B, Wong M-F, Abbeel P, Koller D. Link prediction in relational data. Adv Neural Inf Process Syst 2003;16.
[24] Yu K, Chu W, Yu S, Tresp V, Xu Z. Stochastic relational models for discriminative link prediction. Adv Neural Inf Process Syst 2006;19.
[25] Abid F, Hamami L. A survey of neural network based automated systems for human chromosome classification. Artif Intell Rev 2018;49:41–56.
[26] Liu Y, Xu C, Chen L, Yan M, Zhao W, Guan Z. TABLE: Time-aware Balanced Multi-view Learning for stock ranking. Knowledge-Based Systems 2024;112424.
[27] Ogwueleka FN, Misra S, Colomo-Palacios R, Fernandez L. Neural network and classification approach in identifying customer behavior in the banking sector: A case study of an international bank. Hum Factors Ergon Manuf Serv Ind 2015;25(1):28–42.
[28] Tang Y, Xiao X, Shi Y. Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data. 2014, p. 75–86.
[29] Bouyer A, Beni HA, Arasteh B, Aghaee Z, Ghanbarzadeh R. Fip: A fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks. Expert Syst Appl 2023;213:118869.
[30] Zhuang H, Sun Y, Tang J, Zhang J, Sun X. Influence maximization in dynamic social networks. In: 2013 IEEE 13th international conference on data mining. IEEE; 2013, p. 1313–8.
[31] Wang Y, Fan Q, Li Y, Tan K-L. Real-time influence maximization on dynamic social streams. 2017, arXiv preprint arXiv:1702.01586.
[32] Peng B. Dynamic influence maximization. Adv Neural Inf Process Syst 2021;34:10718–31.
[33] Karp RM. Reducibility among combinatorial problems. In: Complexity of computer computations. 1972, p. 85–103.
[34] Huang S, Bao Z, Culpepper JS, Zhang B. Finding temporal influential users over evolving social networks. In: 2019 IEEE 35th international conference on data engineering. ICDE, IEEE; 2019, p. 398–409.
[35] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. 2010, p. 1029–38.