Information Processing and Management 61 (2024) 103683

Contents lists available at ScienceDirect

Information Processing and Management

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

Influence maximization on hypergraphs via multi-hop influence estimation

Xulu Gong [a], Hanchen Wang [b,*], Xiaoyang Wang [c], Chen Chen [d], Wenjie Zhang [c], Ying Zhang [a,b]

[a] Zhejiang Gongshang University, Hangzhou, 310014, China
[b] University of Technology Sydney, Ultimo, 2007, NSW, Australia
[c] University of New South Wales, Kensington, 2052, NSW, Australia
[d] University of Wollongong, Wollongong, 2522, NSW, Australia

ARTICLE INFO

Keywords:
Influence maximization
Hypergraphs
Social networks
Data mining

ABSTRACT

Influence Maximization (IM) has promising applications in social network marketing and has been extensively researched over the past years. However, previous IM studies mainly focus on ordinary graphs rather than hypergraphs, where edges cannot accurately describe group interactions or relationships. To model group interactions, we investigate the IM problem on hypergraphs under the Susceptible–Infected spreading model with Contact Process dynamics (SICP) in this paper. In this paper, we proposed a probability distribution-based method, called Multi-hop Influence Estimation (MIE), which can accurately estimate the rank of influence expectation of nodes, to solve the IM problem on hypergraphs. Specifically, we compute the influence score for each node through a constrained Depth First Search (DFS) under a probability model, and then select seed node according to the influence score. In addition, by analysing the characteristics of the influence diffusion model, we find that the influence of a node is significantly related to its neighbourhood structure. Based on the observation, we propose a term named neighbourhood coefficient to describe the neighbourhood structure of a node. Further, an efficient and effective method, called Adaptive Neighbourhood Coefficient Algorithm (*Adeff*), is proposed to solve the IM problem on hypergraphs. Extensive experiments on real-world datasets demonstrate the effectiveness and efficiency of our proposed methods. Compared with the state-of-the-art approach, our proposed methods can achieve up to 450% improvement in terms of effectiveness.

1. Introduction

In recent years graph data has a wide range of applications in social networks and data mining (Fan et al., 2010; Li, Qin, et al., 2020; Ma et al., 2020), of which Influence Maximization (IM) is a popular one. IM aims to find a set of individuals who can maximize the influence spread under a certain diffusion model in a social network. This problem is proved to be NP-hard and has a wide range of applications in social networks, *e.g.,* virtual marketing, and has been extensively studied (Cai et al., 2020; Chen et al., 2015; Chen, Wang, & Yang, 2009; Gomez-Rodriguez, Song, Du, Zha, & Schölkopf, 2016; Kumar, Mallik, Khetarpal, & Panda, 2022; Li, Bhowmick, Cui, Gao, & Ma, 2015; Li, Cai, et al., 2020; Morone & Makse, 2015; Wang, Zhang, Zhang, & Lin, 2016a, 2016b; Wang, Zhang, Zhang, Lin & and Chen, 2016; Yan, Huang, Gao, Lu, & He, 2017).

* Corresponding author.
E-mail addresses: 21020100006@pop.zjgsu.edu.cn (X. Gong), hanchen.wang@uts.edu.au (H. Wang), xiaoyang.wang1@unsw.edu.au (X. Wang), chenc@zjgsu.edu.cn (C. Chen), wenjie.zhang@unsw.edu.au (W. Zhang), ying.zhang@uts.edu.au (Y. Zhang).

https://doi.org/10.1016/j.ipm.2024.103683
Received 26 July 2023; Received in revised form 28 December 2023; Accepted 16 January 2024
Available online 15 February 2024

However, existing studies mainly focus on ordinary graphs, whose interactions are only between pairwise individuals. While in some real-world scenarios, interactions or relationships among individuals may exist in a groupwise format instead of a pairwise format. For example, in the co-authorship network where the node represents the author, multiple researchers may contribute to one publication commonly. In such scenarios, one edge in an ordinary graph can only model the co-authorship between two researchers and thus fails to capture the group interactions among all authors in a publication. Hypergraphs are able to capture high-order interactions among individuals where one hyperedge can connect more than two entities, and it is beneficial for modelling the groupwise interactions or relationships among multiple individuals. Therefore, the study of IM on hypergraphs is of great importance. Nevertheless, only a few works focus on this problem which has not been explored sufficiently yet. Hypergraphs have distinguished structural features compared with ordinary graphs, thus the widely used influence diffusion model on original graphs is not applicable on hypergraphs, *e.g.,* the IC model and LT model. Further, methods used in ordinary graphs cannot be adapted directly into hypergraphs, thus leaving the IM problem on hypergraphs as a challenging issue.

In this paper, we focus on the IM problem on hypergraphs under the Susceptible–Infected spreading model with Contact Process dynamics (SICP) model. In the SICP model, at each time step, each node in the infected state will attempt to diffuse its information to its neighbours through the hyperedges it belongs to. Specifically, infected node v will first select one of its hyperedges it belongs to, then attempt to infect each susceptible node on this hyperedge with a given infection probability β. This diffusion process terminates when a given time step threshold T is reached. The purpose of this problem is to detect a set of individuals (seed nodes) with size less than a given constraint K, such that the number of nodes in infected state is maximized when the diffusion process ends. This problem is first studied by Xie et al. and is NP-hard (Xie, Zhan, Liu & Zhang, 2023). Xie et al. assumed that node with larger degree has larger influence and developed a method, HADP namely, which selects node with the maximal degree as the seed node adaptively. However, since the high complexity of the SICP model, degree cannot be used as a metric of node's influence, in other words, the assumption of node with higher degree tends to have higher influence cannot hold (More detailed analysis is presented in Section 4.2.1). Thus HADP failed to detect high quality seed node.

Our solutions. In this paper, we designed a novel probability distribution based approach and a neighbourhood coefficient based approach to solve the IM problem on hypergraphs under the SICP model. In this diffusion model, there are two important factors, *i.e.,* the infection probability β and time step threshold T. These two factors determine the information diffusion process of an infected node, especially T determines the depth of information diffusion. It is intuitive that the optimal seed node sets are different under different combinations of β and T. Therefore, an effective method is supposed to take β and T into account when detecting seed node set. To this end, we consider the diffusion process to follow a probability distribution, and the distribution is specific when β and T are both given. Further, the influence expectation of a node is specific and can be worked out. Thus we can develop a method to calculate the influence expectation for each node and then select seed node according to the rank of influence expectation. However, this method is not practical because of its high time complexity. Therefore, we proposed an efficient and effective method to estimate the rank of influence expectation, which can take β and T into account jointly, namely **Multi-hop Influence Estimation (MIE)**. Specifically, we first analysed the probability diffusion of the SICP model. Based on our analysis we transformed the hypergraph under the SICP model into a directed weighted graph under a SICP-variant diffusion model, where the weight on each edge represents the corresponding infection probability. We use a constrained Depth First Search (DFS) to estimate the rank of influence expectation for all nodes based on probability distribution, which is used for the downstream seed node selection. On the other hand, we analysed the properties and characteristics of the SICP model and proposed a neighbourhood coefficient based approach for this problem, namely **Adaptive Neighbourhood Coefficient Algorithm (*Adeff*)**. Specifically, we found that node with more neighbours on each hyperedge tends to have higher influence. We defined a new term named neighbourhood coefficient to describe the characteristics of node's neighbourhood structure, and higher neighbourhood coefficient indicates more average neighbours on each hyperedge for a node. The seed node is selected with the neighbourhood coefficients for the nodes. Furthermore, to alleviate the influence overlap among seed nodes and increase the overall influence, we propose an effective pruning method.

We conduct experimental study to compare our proposed methods with the newest approach currently, *i.e.,* HADP (Xie, Zhan, Liu & Zhang, 2023), in terms of effectiveness and efficiency. It is shown that our proposed algorithms improve the effectiveness by up to 450% compared with HADP and achieve new state-of-the-art performance.

Contributions. We summarized the contributions of this paper as follows:

- We proposed a method to transform the hypergraph under SICP model into a directed weighted graph under a SICP-variant model. We proved node has the same influence expectation under the SICP model and SICP-variant model. Thus, the IM problem on hypergraphs under SICP model can be transformed into the problem of IM on directed weighted graphs under the SICP-variant model, which is easier to handle.
- We proposed a probability distribution-based method, namely Multi-hop Influence Estimation (MIE), which iteratively selects the seed nodes according to the influence expectation for each node computed by a constrained DFS considering the infection probability β and time step threshold T.
- We design a novel neighbourhood coefficient that captures the key structural information of the hypergraph, based on which we develop an efficient and effective method named Adaptive Neighbourhood Coefficient Algorithm (*Adeff*) to solve the IM problem on hypergraphs.
- Extensive experiments on 8 real-world datasets demonstrate the superiority (up to 4.5×) of our proposed algorithms compared with existing methods.

Table 1
Notations.

Notations	Description
$H(V, E)$	An hypergraph with node set V and hyperedge set E
$N(v, H)$	The neighbours of v in H
$\|e_i\|$	The number of nodes hyperedge e_i contains
E_v	The set of hyperedges which contain node v
$deg(v)$	The degree of node v
$Hdeg(v)$	The hyperdegree of node v
$\mathcal{M}$	The influence diffusion model
S	A selected seed node set $S \subseteq A$
$\sigma_H(S)/\sigma(S)$	The influence of seed node set S in H
$fl(S)/fl(v)$	The node set infected by seed set S/ seed node v

Organization. The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 introduces the preliminaries, including the definition of hypergraph, the diffusion model we used and the problem definition. Our proposed algorithms MIE and *Adeff* are introduced in Section 4. We report the experimental result in Section 5. Section 6 concludes the paper.

2. Related works

Influence Maximization. The goal of Influence maximization problem is to detect a set of individuals who play import roles in social networks marketing. Kempe, Kleinberg, and Tardos (2003) first proposed this problem in 2003. They propose two widely used diffusion models for this problem, *i.e.,* the independent cascade (IC) and linear threshold (LT) diffusion models and provide a greedy framework with $(1 - 1/e)$ approximation ratio. Borgs, Brautbar, Chayes, and Lucier (2014) propose a near-linear time method, the reverse influence sampling framework, for the IC model. Tang, Xiao, and Shi (2014) propose TIM/TIM+ algorithm to improve the sampling efficiency. Tang, Shi, and Xiao (2015) further improve the sampling efficiency based on martingales. Recently extensive efforts have been made to solve the influence maximization problem (Cai et al., 2020; Kumar et al., 2022; Wang et al., 2016b), and are surveyed in Aghaee, Ghasemi, Beni, Bouyer, and Fatemi (2021), Banerjee, Jenamani, and Pratihar (2020) and Singh, Srivastva, Verma, and Singh (2022).

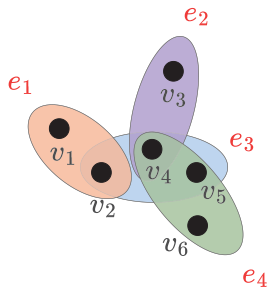
Influence Maximization on Hypergraphs. Modelling social networks as hypergraphs is considered to be able to capture high-order interactions between individuals, however the problem of influence maximization on hypergraphs caught little attentions in recent years. Amato et al. propose to model social media networks by hypergraphs, which can represent "user-to-multimedia" relationships by hyperedges (Amato, Moscato, Picariello, & Sperlí, 2017). Zhu et al. propose the social IM problem in hypergraph under IC model, and prove the NP-hard property of this problem (Zhu, Zhu, Ghosh, Wu, & Yuan, 2018). In addition, authors propose an algorithm for general weighted social influence maximization problem which preserves $(1 - 1/e - \epsilon)$ approximation. Antelmi et al. generalize the LT diffusion model to hypergraphs and design a greedy algorithm based on a subtractive approach, and they adopt a pruning strategy to remove unnecessary nodes or edges from the original hypergraph (Antelmi, Cordasco, Spagnuolo, & Szufel, 2021). MA and Rajkumar (2022) study the IM problem on hypergraph under the HyperCascade diffusion model and generalize a ranking based algorithm for IM problem on ordinary graphs, IMRANK (Cheng, Shen, Huang, Chen, & Cheng, 2014) to hypergraphs. Aktas et al. study the influential hyperedge detection problem, which aims to detect critical, hyperedges (Aktas, Jawaid, Gokalp, & Akbas, 2022). Su et al. study the IM problem on hypergraphs with a novel causal objective (Su & Zhang, 2023). Authors consider each individual carries specific attributes with individual treatment effect (ITE), and the sum of ITEs of the infected is a more reasonable objective for influence spread. Besides, authors develop an algorithm, CauIM, to solve this problem, which can extract approximately optimal seed sets. Xie et al. is the first work of studying the problem of IM on hypergraphs under the SICP model and they propose a degree-based heuristic *i.e.,* HADP, to solve this problem (Xie, Zhan, Liu & Zhang, 2023). However, degree cannot model the complex hypergraph structure and influence diffusion process well, thus HADP fails to detect seed nodes with high influence.

3. Preliminaries

In this paper, we focus on the influence maximization problem on hypergraphs. In this section, we introduce the preliminaries. The frequently used notations are summarized in Table 1.

3.1. Hypergraph

Let $H(V, E)$ denote a hypergraph, V and E denote set of nodes and hyperedges respectively, $N = |V|$ and $M = |E|$ denote the number of nodes and hyperedges of H respectively. Different from ordinary graphs, in a hypergraph H each hyperedge e_i contains



(a) Hypergraph H

	e_1	e_2	e_3	e_4
v_1	1	0	0	0
v_2	1	0	1	0
v_3	0	1	0	0
v_4	0	1	1	1
v_5	0	0	1	1
v_6	0	0	0	1

(b) The incidence matrix of H

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	0	0
v_2	1	0	0	1	1	0
v_3	0	0	0	1	0	0
v_4	0	1	1	0	2	1
v_5	0	1	0	2	0	1
v_6	0	0	0	1	1	0

(c) The adjacency matrix of H

Fig. 1. An example of hypergraph.

two or more nodes. We use $|e_i|$ to denote the number of nodes e_i contains. We use $E_v = \{e_i \in E | v \in e_i\}$ to denote the set of hyperedges which contains node v. The hypergraph H can be denoted by an incidence matrix $C \in \mathbb{R}^{|V|\times|E|}$, where

$$C_{ik} = \begin{cases} 1 & \text{if } i \in e_k \\ 0 & \text{if } i \notin e_k \end{cases} \tag{1}$$

The adjacency matrix of hypergraph H is represented as $A \in \mathbb{R}^{N\times N}$, where A_{ij} is the number of common hyperedges v_i and v_j both belong to. We present an example of hypergraph in Fig. 1.

The **degree** of a node v_i ($deg(v_i)$) is defined as the number of neighbours v_i has, and node u is the neighbour of node v iff there exists at least one hyperedge which u and v both belong to. It can be formulated as follows:

$$deg(v_i) = \sum_{j=1}^{N} \mathcal{A}_{ij}, \tag{2}$$

where $\mathcal{A} \in \mathbb{R}^{n\times n}$, $\mathcal{A}_{ij} = 1$ if $A_{ij} > 0$ and $\mathcal{A}_{ij} = 0$ else.

We use $Hdeg(v_i)$ to denote the **hyperdegree** of v_i. It means the number of hyperedges which contain v_i and can be computed as follows:

$$Hdeg(v_i) = \sum_{j=1}^{M} C_{ij}. \tag{3}$$

For example, the degree of node v_4 in Fig. 1(a) is $deg(v_4) = 4$, and the hyperdegree of v_4 is $Hdeg(v_4) = 3$.

3.2. Diffusion model

In this work, we use the **Susceptible–Infected spreading model with Contact Process dynamics** (SICP) (Xie, Zhan, Liu & Zhang, 2023) as the diffusion model to quantify the influence of a set of seed nodes. In this diffusion model, there exists two node states, *i.e.,* susceptible (S) state or infected (I) state, and each node can only be in one of them at any time step. The influence diffusion process on hypergraphs of a seed set S can be described as:

- **Step 1:** In the beginning, all nodes in seed node set S are in I state, and the remaining nodes are in S state.
- **Step 2:** At each time step t, each infected node v_i will first randomly select one hyperedge e_k it belongs to. For each susceptible node v_j in e_k, v_i attempts to infect it with infection probability β.
- **Step 3:** Repeat Step 2 until t reaches a preset value T, where T is a control parameter.

3.3. Problem definition

Definition 1 (*Influence*)**.** Given a hypergraph $H(V, E)$, a seed set $S \subseteq V$, and a diffusion model $\mathcal{M}$. The influence of S, represented by $\sigma_H(S)$, is defined as the expected number of infected nodes in H at the end of the diffusion process when only the nodes in S are infected initially, and we call the node set infected by S **followers**, denoted by $fl(S)$, where $|fl(S)| = \sigma_H(S)$. When the context is clear, for simplicity we also use $\sigma(S)$ to denote $\sigma_H(S)$.

Definition 2 (*Influence Maximization on Hypergraph*)**.** Given a hypergraph $H(V, E)$, an positive integer K for the size of seed set and a diffusion model $\mathcal{M}$, the problem of influence maximization on hypergraphs is to find a seed set S^* of K nodes in H, such that the expected number of infected nodes is maximized:

$$S^* = \underset{S\subseteq V \wedge |S|=K}{\arg\max}\ \sigma(S)$$

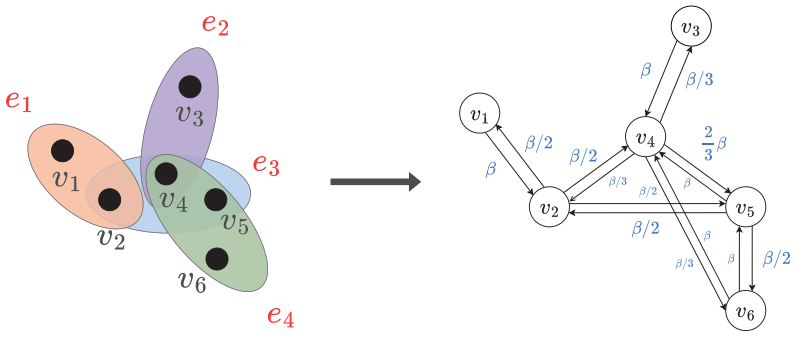


Fig. 2. An example of equivalent probability transformation.

In this paper, we study the IM problem on hypergraph (HyperIM) when $\mathcal{M}$ is set as the SICP model, and this problem is NP-hard (Xie, Zhan, Liu & Zhang, 2023; Zhu et al., 2018).

4. Algorithms

In this section, we introduce the algorithms we proposed, *i.e.,* the Multi-hop Influence Estimation (MIE) and Adaptive Neighbourhood Coefficient Algorithm (*Adeff*).

4.1. Multi-hop influence estimation

In this subsection we introduce the MIE algorithms. Specifically, we first transform the hypergraph under the SICP model into a directed weighted graph under a SICP-variant diffusion model. Then we estimate the rank of influence expectation for all nodes based on probability distribution with a constrained DFS. Finally, we select seed nodes iteratively according to the estimated rank of influence expectation, and the rank is updated in each iteration.

4.1.1. Hypergraph to directed weighted graph

In this subsection, we discuss to transform the HyperIM problem under the SICP diffusion model to a directed weighted graph IM problem under a SICP-like diffusion model by equivalent probability transformation. From the diffusion process of SICP model, we can find that the diffusion of this model is highly stochastic. One of the key factors causing the high diffusion stochasticity is the random selection to the hyperedge of a infected node at each time step t. A infected node v may belong to more than one hyperedge simultaneously, *i.e.,* $|E_v| > 1$, and different hyperedges $e \in E_v$ may contain significantly different neighbourhood structure (including number of neighbours and neighbours' topological features), which means that different hyperedge selections have different influence diffusion results. Therefore, it is important to take the stochasticity of hyperedge selection into account when selecting a seed node.

To better describe and handle the stochasticity of hyperedge selection, we propose to transform the selection of hyperedge to selection on nodes according to an equivalent probability. To explain the **equivalent probability transformation**, we take the hypergraph H in Fig. 1 as an example. Node v_4 belongs to 3 hyperedges (e_2, e_3, e_4), it has 4 neighbours (v_2, v_3, v_5, v_6). Initially only v_4 is infected. If v_4 wants to infect v_3 by at step one, it has to select hyperedge e_2 in the first phase (with probability $1/3$) and then infect v_3 with probability β. Thus the probability of v_4 infects v_3 by one time step is $1/3 * \beta$. See another example of v_4 infects v_5: as v_4 and v_5 have 2 common hyperedges (e_3, e_4), the probability of v_4 infects v_5 via hyperedge e_3 and e_4 are both $\beta/3$, thus the probability of v_4 infects v_5 by time step one is $2/3 * \beta$ (see Fig. 2). We formulate the definition of **equivalent probability transformation** as follows:

Definition 3 (*Equivalent Probability Transformation*)**.** Given a hypergraph $H(V, E)$, the equivalent probability transformation on H is to transform H to a directed weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = V$ is the node set, $\mathrm{v_i} \in \mathcal{V}$ is the transformed version of node $v_i \in V$, $\mathcal{E}$ is the directed edge set, $\mathcal{E}_{ij} = < v_i, v_j >$ is a directed edge from node v_i to node v_j, $\mathcal{E}_{ij} \in \mathcal{E}$ iff there exists at least one hyperedge $e \in E$ and e contains both v_i and v_j, $\mathcal{W}$ is the weight matrix and $\mathcal{W}_{ij} \in \mathcal{W}$ is the weight on directed edge $< v_i, v_j >$. The $\mathcal{W}_{ij}$ is computed by follows:

$$\mathcal{W}_{ij} = \frac{|E_i \cap E_j|}{|E_i|}\beta, \tag{4}$$

where E_i and E_j denote the set of hyperedges which v_i and v_j belongs to respectively.

By the equivalent probability transformation we can transform the hypergraph $H(V,E)$ under the SICP diffusion model to a directed weighted graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$, thus the stochasticity of hyperedge selection is denoted by infection probability distribution over nodes. Then we need to define a new diffusion model applying on $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$, which is similar with the SICP model. We denote the new diffusion model as SICP-variant, and it works as follows:

- **Step 1:** In the beginning, all nodes in S are in I state, and the remaining nodes are in S state.
- **Step 2:** At each time step t, each infected node v_i will attempt to infect each of its susceptible neighbour v_j with infection probability $\mathcal{W}_{ij}$.
- **Step 3:** Repeat Step 2 until t reaches a preset T.

Lemma 1. *Given a hypergraph $H(V,E)$, number of independent repeated experiments $\theta \in \mathbb{N}$, infection probability β, $\{v_i, v_j\} \in V$ is an adjacent node pair. Initially only node v_i is infected, in each independent experiment, v_i will first randomly select one hyperedge e_k it belongs to, then for each susceptible node u in e_k, v_i attempts to infect it with probability β. Conducting the independent experiments above θ times, $cnt_\theta(v_i \to v_j)$ denotes the number of independent experiments where v_i infected v_j successfully during the θ times of independent repeated experiments. Then:*

$$\lim_{\theta\to+\infty} \frac{cnt_\theta(v_i \to v_j)}{\theta} = \frac{|E_i \cap E_j|}{|E_i|}\beta, \tag{5}$$

where E_i denotes the set of hyperedges v_i belongs to.

Proof. Let $m = |E_i|$, $n = |E_i \cap E_j| (n \geq 1)$ and $\{e'_1, e'_2, \ldots, e'_n\} = E_i \cap E_j$ denotes the set of hyperedges v_i and v_j both belong to. In one independent experiment, for each hyperedge $e'_k \in E_i \cap E_j$, the probability of v_i select e'_k is $\frac{1}{m}$, then for susceptible node v_j in e'_k, the probability of v_i infects v_j by e'_k successfully is $\frac{1}{m}\beta$, *i.e.,*

$$\lim_{\theta\to+\infty} \frac{cnt_\theta(v_i \to v_j | e'_k)}{\theta} = \frac{1}{m}\beta,$$

where $cnt_\theta(v_i \to v_j | e'_k)$ represents the number of independent experiments where v_i infected v_j by e'_k successfully during the θ times of independent repeated experiments. Further,

$$\begin{aligned}
\lim_{\theta\to+\infty} \frac{cnt_\theta(v_i \to v_j)}{\theta} &= \sum_{k=1}^{n} \lim_{\theta\to+\infty} \frac{cnt_\theta(v_i \to v_j | e'_k)}{\theta} \\
&= \sum_{k=1}^{n} \frac{1}{m}\beta \\
&= \frac{n}{m}\beta \\
&= \frac{|E_i \cap E_j|}{|E_i|}\beta \quad \square
\end{aligned}$$

Theorem 1. *Given a hypergraph $H(V,E)$, rational number $\beta \in [0,1]$ and positive integer T for the SICP diffusion model, $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ is a directed weighted graph obtained by the equivalent probability transformation on H described on Definition 3, for node $v_i \in V$ and node $\mathsf{v}_i \in \mathcal{V}$, $\mathbb{E}(\sigma_H(v_i))$ denote the expectation of the influence of v_i under the SICP diffusion model and $\mathbb{E}(\sigma_\mathcal{G}(\mathsf{v}_i))$ denote the expectation of the influence of v_i under the SICP-variant diffusion model. Then, $\mathbb{E}(\sigma_H(v_i)) = \mathbb{E}(\sigma_\mathcal{G}(\mathsf{v}_i))$.*

Proof. $\mathbb{E}(\sigma_H(v_i)) = \lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_H^{(k)}(v_i)}{\theta}$ and $\mathbb{E}(\sigma_\mathcal{G}(\mathsf{v}_i)) = \lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_\mathcal{G}^{(k)}(\mathsf{v}_i)}{\theta}$, $\theta \in \mathbb{Z}^+$, $\sigma_H^{(k)}(v_i)$ and $\sigma_\mathcal{G}^{(k)}(\mathsf{v}_i)$ denotes the number of infected nodes in kth diffusion process under SICP model and SICP-variant model with only node v_i and v_i is infected initially respectively. For each node $v_j \in V$ satisfying the shortest path length between v_j and v_i is no more than T, $\exists\ p_{ij} \in \mathbb{Q}$ and $p_{ij} \in (0,1]$, s.t., p_{ij} is the probability of v_i infected v_j under the SICP model, where $\mathbb{Q}$ represents the rational number set. During the θ times of diffusion process, the expectation of the times of v_j infected by v_i is $p_{ij} \cdot \theta$, when $\theta \to +\infty$, $p_{ij} \cdot \theta \to +\infty$. Thus, for each node $v_l \in V$ satisfying the shortest path length between v_l and v_i is no more than $T-1$, when $\theta \to +\infty$, the number of times v_i infected v_l goes to $+\infty$, and the number of times v_l attempt to infect each of its 1-hop neighbours under SICP model within one time step (denoted by θ') goes to $+\infty$ also. From Lemma 1, $\lim_{\theta'\to+\infty} \frac{cnt_{\theta'}(v_l \to v_p)}{\theta'} = \frac{|E_l \cap E_p|}{|E_l|}\beta$, where $v_p \in N(v_l, H)$. Therefore, when $\theta \to +\infty$, for each node $v_l \in V$ satisfying the shortest path length between v_l and v_i is no more than $T-1$, for each node $v_p \in N(v_l, H)$, the probability of v_l infects v_p within one time step is equal to $\frac{|E_l \cap E_p|}{|E_l|}\beta$. Thus $\lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_{H^T(v_i)}^{(k)}(v_i)}{\theta} = \lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_{\mathcal{G}^T(\mathsf{v}_i)}^{(k)}(\mathsf{v}_i)}{\theta}$ holds, where $H^T(v_i)$ is the subgraph of H by removing each node in H whose shortest path length away from v_i is larger than T, and the same with $\mathcal{G}^T(\mathsf{v}_i)$. Under the time step threshold T, $\sigma_H^{(k)}(v_i) = \sigma_{H^T(v_i)}^{(k)}(v_i)$ and $\sigma_\mathcal{G}^{(k)}(\mathsf{v}_i) = \sigma_{\mathcal{G}^T(\mathsf{v}_i)}^{(k)}(v_i)$ hold. Hence, $\lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_H^{(k)}(v_i)}{\theta} = \lim_{\theta\to+\infty} \frac{\sum_{k=1}^{\theta} \sigma_\mathcal{G}^{(k)}(\mathsf{v}_i)}{\theta}$ holds. Further, $\mathbb{E}(\sigma_H(v_i)) = \mathbb{E}(\sigma_\mathcal{G}(\mathsf{v}_i))$, Theorem 1 holds. $\square$

By Theorem 1 we can know that the expectation of the influence of a node $v \in V$ keep unchanged after the equivalent probability transformation. Hence the node with the maximal influence on $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ under the SICP-variant model also has the maximal influence on $H(V,E)$ under the SICP model, and we can develop methods applying on $\mathcal{G}$ to find the node with the maximal influence on H indirectly and then obtain the seed node set incrementally.

4.1.2. Motivation and basic algorithm framework

As discussed in sub Section 4.1.1, a hypergraph $H(V, E)$ under the SICP model can be transformed to a directed weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ under the SICP-variant model. Intuitively, the influence of a node $\mathsf{v} \in \mathcal{G}$, $\sigma(\mathsf{v})$, is a random variable and follows a certain probability distribution $\mathcal{P}(x)$. Theoretically, $\mathcal{P}(x)$ is explicit when β and T is given explicitly. Further the expectation of $\sigma(\mathsf{v})$ can be computed exactly. We formulate such claim as Theorem 2.

Theorem 2. *Given a directed weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, rational number $\beta \in [0, 1]$ and positive integer T for the SICP-variant diffusion model, for node $\mathsf{v} \in \mathcal{V}$, $\mathbb{E}(\sigma(\mathsf{v}))$ is the expectation of $\sigma(\mathsf{v})$. Then, $\exists\ \Omega \in \mathbb{R}$ s.t. $\mathbb{E}(\sigma(\mathsf{v})) = \Omega$.*

Motivated by this theorem, we develop an algorithm to compute $\mathbb{E}(\sigma(\mathsf{v}))$ exactly. Considering the high time complexity of exact algorithm, we propose a method to estimate the rank of $\mathbb{E}(\sigma(\mathsf{v}))$ with high accuracy and acceptable time consuming. We summarize the general framework of our proposed algorithm as Algorithm 1, and we name it as **Multi-hop Influence Estimation (MIE)**.

Algorithm 1 Basic Algorithm Framework (MIE)

Input: Hypergraph $H(V, E)$, infection probability β, time step threshold T, seed set size K, number of simulations θ, path length constraint L.
Output: Set of seed node S.

$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W}) \leftarrow$ equivalent probability transformation on $H(V, E)$
$S = \emptyset$
$\mathcal{G}^{(0)} \leftarrow \mathcal{G}$
while $|S| < K$ **do**
  $k \leftarrow |S|$
  $\Delta[\cdot] \leftarrow$ MultihopInfluenceScore$(\mathcal{G}^{(k)}, \beta, T, \mathsf{L})$
  $s \leftarrow \arg\max_{\mathsf{v} \in \mathcal{V}^{(k)}} \Delta[\mathsf{v}]$
  $S \leftarrow S \cup \{s\}$
  $\mathcal{G}^{(k+1)} \leftarrow$ InfluenceOverlapAlleviation$(\mathcal{G}^{(k)}, H, s, \theta)$
end while
return S

4.1.3. Theoretical computation of influence expectation

In graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, for a node $\mathsf{v}_\mathsf{i} \in \mathcal{V}$, its influence expectation can be computed as the sum of probability v_i infects each of the rest nodes in $\mathcal{V}$. Let only node v_i is infected initially, and $\mathcal{P}_T(\mathsf{i} \to \mathsf{j})$ denotes the probability of v_j is in infected state at time step T under the SICP-variant model, then

$$\mathbb{E}(\sigma(\mathsf{v}_\mathsf{i})) = \sum_{\mathsf{v}_\mathsf{j} \neq \mathsf{v}_\mathsf{i}, \mathsf{v}_\mathsf{j} \in \mathcal{V}} \mathcal{P}_T(\mathsf{i} \to \mathsf{j}). \tag{6}$$

Note that $\mathcal{P}_T(\mathsf{i} \to \mathsf{j}) = 0$ for each node v_j which does not lie within the T-hop neighbourhood of v_i. Given path length constraint T, it exists multiple paths from v_i to v_j (multi-path infection). Using $\mathsf{path}_{\mathsf{i} \to \mathsf{j}} = \{\mathsf{pt}_1, \mathsf{pt}_2, \ldots, \mathsf{pt}_L\}$ to denote all the paths from v_i to v_j where each path's length is no more than T and does not contain circle. Using $\mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l)$ to denote the probability of v_j is in infected state at time step T and is infected by v_i through path pt_l. Then,

$$\mathcal{P}_T(\mathsf{i} \to \mathsf{j}) = 1 - \prod_{\mathsf{pt}_l \in \mathsf{path}_{\mathsf{i} \to \mathsf{j}}} (1 - \mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l)) \tag{7}$$

We explain Eq. (7) by an example illustrated in Fig. 3. There exists multiple paths from v_i to v_j. For each path pt_l, the probability of v_j is in infected state at time step T and is infected by v_i through path pt_l is $\mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l)$, then the probability of v_j is not infected by v_i through pt_l when time step reaches T is $1 - \mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l)$. Further, the probability of v_j is not in infected state at time step T is the product of probability of each path fails to infect v_j when time step reaches T, *i.e.,* $\prod_{\mathsf{pt}_l \in \mathsf{path}_{\mathsf{i} \to \mathsf{j}}} (1 - \mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l))$. Further we have Eq. (7).

4.1.4. Computation of the infection probability over one path

In the subsection, we discuss how to compute the infection probability over one path, *i.e.,* $\mathcal{P}_T(\mathsf{i} \to \mathsf{j} \mid \mathsf{pt}_l)$ in Eq. (7). In Fig. 4 we illustrate several infection paths with different lengths, $\beta_i \in [0, 1)$ on the directed edge represents corresponding infection probability between two nodes. Initially, only node 1 is in infected state. We use $\widetilde{\mathcal{P}}_T(n)$ to represent the probability of node n is in infected state at time step T. We use $\mathcal{P}_n(j)$ to represent the probability of node n is not in infected state before time step j and is infected at time step j. We discuss the computation of infection probability under time step threshold T in the 4 cases respectively:

- Case 1 (Fig. 4(a)): the path length between source node and target node is 1. The probability of node 2 is not in infected state in time step T is $(1 - \beta_1)^T$, then $\widetilde{\mathcal{P}}_T(2) = 1 - (1 - \beta_1)^T$.

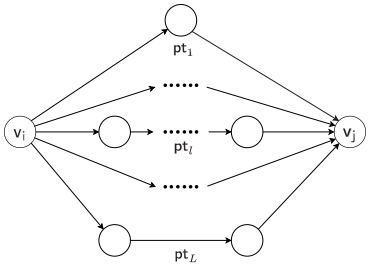


Fig. 3. An example of multi-path infection.

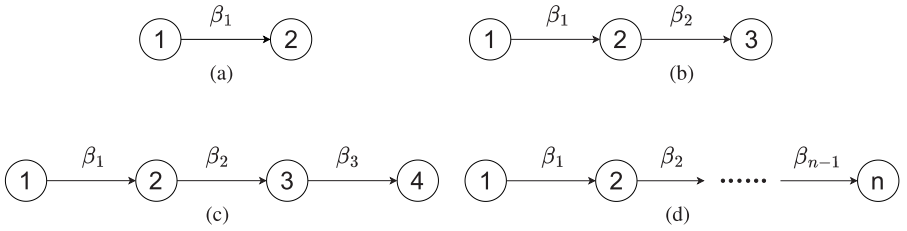


Fig. 4. Infection paths with different length.

- Case 2 (Fig. 4(b)): the path length between source node and target node is 2. Then,

$$\widetilde{\mathcal{P}}_T(3) = \sum_{i=2}^{T} \mathcal{P}_3(i). \tag{8}$$

Assuming j ($j < i$) is the time step when node 2 is infected, in this scenario the probability of node 3 is infected at time step i (is not in infected state before time step i) is $\mathcal{P}_2(j)(1-\beta_2)^{i-1-j}\beta_2$. Then,

$$\mathcal{P}_3(i) = \sum_{j=1}^{i-1} \mathcal{P}_2(j)(1-\beta_2)^{i-1-j}\beta_2 \tag{9}$$

Specifically, $\mathcal{P}_2(j) = (1-\beta_1)^{j-1}\beta_1$. Then, we have

$$\begin{aligned}\widetilde{\mathcal{P}}_T(3) &= \sum_{i=2}^{T} \mathcal{P}_3(i) \\ &= \sum_{i=2}^{T}\sum_{j=1}^{i-1} \mathcal{P}_2(j)(1-\beta_2)^{i-1-j}\beta_2 \\ &= \sum_{i=2}^{T}\sum_{j=1}^{i-1} (1-\beta_1)^{j-1}\beta_1(1-\beta_2)^{i-1-j}\beta_2\end{aligned} \tag{10}$$

- Case 3 (Fig. 4(c)): the path length between source node and target node is 3. Similar with case 2, we have

$$\widetilde{\mathcal{P}}_T(4) = \sum_{i=3}^{T} \mathcal{P}_4(i). \tag{11}$$

Assuming j ($2 \le j < i$) is the time step when node 3 is infected, in this scenario the probability of node 4 is infected at time step i (is not in infected state before time step i) is $\mathcal{P}_3(j)(1-\beta_3)^{i-1-j}\beta_3$. Then,

$$\mathcal{P}_4(i) = \sum_{j=2}^{i-1} \mathcal{P}_3(j)(1-\beta_3)^{i-1-j}\beta_3. \tag{12}$$

Algorithm 2 MultihopInfluenceScore

Input: Directed weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, infection probability β, time step threshold T, path length constraint L.
Output: L-hop Influence score $\Delta[\cdot]$.

$\Delta[\cdot] \leftarrow 0$
for each node $\mathsf{v_i} \in \mathcal{V}$ **do**
  $\mathsf{path_{i\rightarrow}} \leftarrow$ ConstrainedDFS($\mathcal{G}, \mathsf{v_i}, \mathsf{L}$) //Searching for all paths starting from $\mathsf{v_i}$ that do not exceed L in path length and do not contain loops
  $\mathcal{V}_{end} \leftarrow$ All end nodes in $\mathsf{path_{i\rightarrow}}$.
  for each node $\mathsf{v_j} \in \mathcal{V}_{end}$ **do**
    $\mathcal{P}_T(\mathsf{i}\rightarrow\mathsf{j}) \leftarrow$ Estimate the probability of $\mathsf{v_j}$ is in infected state at time step T by Eqs. (7), (16) and (17)
    $\Delta[\mathsf{i}] \leftarrow \Delta[\mathsf{i}] + \mathcal{P}_T(\mathsf{i}\rightarrow\mathsf{j})$
  end for
end for
return $\Delta[\cdot]$

Therefore, we have

$$\begin{aligned}\widetilde{\mathcal{P}}_T(4) &= \sum_{i=3}^{T} \mathcal{P}_4(i) \\ &= \sum_{i=3}^{T}\sum_{j=2}^{i-1} \mathcal{P}_3(j)(1-\beta_3)^{i-1-j}\beta_3 \\ &= \sum_{i=3}^{T}\sum_{j=2}^{i-1}\sum_{k=1}^{j-1} \mathcal{P}_2(k)(1-\beta_2)^{j-1-k}\beta_2(1-\beta_3)^{i-1-j}\beta_3 \\ &= \sum_{i=3}^{T}\sum_{j=2}^{i-1}\sum_{k=1}^{j-1} (1-\beta_1)^{k-1}\beta_1(1-\beta_2)^{j-1-k}\beta_2(1-\beta_3)^{i-1-j}\beta_3\end{aligned} \tag{13}$$

- Case 4 (Fig. 4(d)): the path length between source node and target node is n $(2 \le n \le T)$. Similarly, we have

$$\widetilde{\mathcal{P}}_T(n) = \sum_{i=n-1}^{T} \mathcal{P}_n(i). \tag{14}$$

Similar with Eqs. (9) and (12), we have

$$\mathcal{P}_n(i) = \sum_{j=n-2}^{i-1} \mathcal{P}_{n-1}(j)(1-\beta_{n-1})^{i-1-j}\beta_{n-1},\ (n \ge 3, i \ge n-1). \tag{15}$$

In summary, we can compute $\mathcal{P}_T(\mathsf{i}\rightarrow\mathsf{j}\,|\,\mathsf{pt}_l)$ as follows. Using $(\beta_1, \beta_2, \ldots ..., \beta_n)$ to denote the infection probability on each of the edges in infection path pt_l respectively, where n is the path length, $n = |\mathsf{pt}_l|$. Then,

$$\mathcal{P}_T(\mathsf{i}\rightarrow\mathsf{j}\,|\,\mathsf{pt}_l) = \sum_{i=n-1}^{T} \mathcal{P}_n(i), \tag{16}$$

and,

$$\mathcal{P}_n(i) = \begin{cases} \sum_{j=n-2}^{i-1} \mathcal{P}_{n-1}(j)(1-\beta_{n-1})^{i-1-j}\beta_{n-1} & n \ge 3 \\ (1-\beta_1)^{i-1}\beta_1 & n = 2. \end{cases} \tag{17}$$

4.1.5. Influence expectation rank estimation

In sub Section 4.1.3 and 4.1.4 we discussed how to compute the influence expectation $\mathbb{E}(\sigma_{\mathcal{G}}(\mathsf{v_i}))$ for node $\mathsf{v_i} \in \mathcal{V}$ theoretically. To compute $\mathbb{E}(\sigma_{\mathcal{G}}(\mathsf{v_i}))$, a critical procedure is to search all the paths from $\mathsf{v_i}$ (source node) to $\mathsf{v_j}$ (target node) where each path's length is no more than T and does not contain circle for each node $\mathsf{v_j}$ lie within the T-hop neighbourhood of $\mathsf{v_i}$. However we cannot find all the paths above with polynomial time complexity. Thus it is necessary to develop a method which can estimate $\mathbb{E}(\sigma_{\mathcal{G}}(\mathsf{v_i}))$ with high accuracy and acceptable time consuming. In fact, for greedy-based approaches we do not need to estimate the influence expectation itself while we only need to estimate the rank of influence expectation such that we can select the node in the top of the rank as a seed node. To this end, we propose an approach to estimate the rank of influence expectation. Specifically, we constrain the infection path length as L by constrained Depth First Search (DFS) when searching all the path from source node to target node. For each infection path, we compute its corresponding infection probability from source node to target node. Then

Algorithm 3 ConstrainedDFS

Input: Directed weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, starting node $v_i \in \mathcal{V}$, path length constraint L.
Output: $path_{i\rightarrow .}$: All paths starting from v_i that do not exceed L in path length and do not contain loops.

$path_{i\rightarrow .} \leftarrow \emptyset$
$stack = [] \leftarrow$ Initial an empty stack; $pt = [] \leftarrow$ Initial an empty stack
pt.*push*(v_i)
END = *False*
while pt *is not empty* **do**
  if !*END* **then**
    if pt.*length*() $\leq \mathsf{L}$ **then**
      $v_j \leftarrow$ pt.*top*()
      stack.*push*('&')
      for each node $v_k \in N(v_j, \mathcal{G})$ and v_k not in pt **do**
        stack.*push*(v_k)
      end for
    else
      pt.*pop*()
    end if
  end if
  v_q = *stack*.*pop*()
  END = *False*
  if v_q != '&' **then**
    pt.*push*(v_q)
    $path_{i\rightarrow .} \leftarrow path_{i\rightarrow .} \cup \{pt\}$
  else
    pt.*pop*()
    END = *True*
  end if
end while
return $path_{i\rightarrow .}$

we estimate the probability that target node is in infected state at time step T by Eqs. (7), (16) and (17) . Finally, the estimated probability of all target nodes starting from the same source node is summed up to represent the influence score of the source node. After the influence scores of all nodes are obtained, we estimate the rank of influence expectation by the rank of influence scores, *i.e.,* we consider the node with the maximal influence score has the maximal influence expectation. The multi-hop influence score computation algorithm is presented in Algorithm 2, and the constrained DFS algorithm is presented in Algorithm 3. Note that $\mathsf{L} \ll T$ and $\Delta[v_i] < \mathbb{E}(\sigma_{\mathcal{G}}(v_i))$ generally for node v_i, if $\mathsf{L} = T$, $\Delta[v_i] = \mathbb{E}(\sigma_{\mathcal{G}}(v_i))$. In practice, we set L as 1,2 and 3.

4.1.6. Influence overlap alleviation

By Algorithm 2 we can compute the influence score for each node, and estimate the rank of influence expectation by the rank of influence score. Intuitively we can develop an approach which selects nodes with the top K maximal influence score $\Delta[\cdot]$ as the seed nodes. However, there would be a problem lying in such approach, *i.e.,* the influence overlap problem. We describe the influence overlap problem as follows, for seed node $u, v \in S$, $\sigma(\{u, v\}) < \sigma(\{u\}) + \sigma(\{v\})$, which means the selected seed nodes u and v have many common followers. While a high-quality seed node set $\{u, v\}$ is expected to satisfy $\sigma(\{u, v\}) \geq \sigma(\{u\}) + \sigma(\{v\})$. To alleviate this problem, we propose to select node with the maximal influence score $\Delta[\cdot]$ adaptively by iteration. Specifically, in each iteration we select the node with the maximal influence score $\Delta[\cdot]$ and add it into the current seed node set, then we remove it and its main followers from current graph. After selecting a seed node s, We conduct θ times of influence diffusion samplings on the original hypergraph H under SICP model. Then we select those nodes which are infected by s the most times during the influence diffusion samplings as the main followers of s, denoted by $\widehat{fl}(s)$. $\Delta[\cdot]$ is recomputed in next iteration. We summarize the influence overlap alleviation algorithm in Algorithm 4.

4.2. Neighbourhood coefficient based heuristic

4.2.1. Analysis

From the diffusion process of SICP model, we can find that the propagation of this model is highly stochastic. Different from general diffusion models for ordinary graphs like the IC model (Li, Fan, Wang, & Tan, 2018), whose diffusion stochasticity is induced only by the activation probabilities on edges, the diffusion stochasticity of SICP model is caused by two factors. One is the stochasticity caused when an infected node v_i randomly choose a hyperedge e_k it belongs to. The another one is the stochasticity

Algorithm 4 InfluenceOverlapAlleviation

Input: Current graph $\mathcal{G}^{(k)}$, hypergraph H, current seed node s, number of simulations θ.
Output: New graph $\mathcal{G}^{(k+1)}$ after influence overlap alleviation.

$fl(s) = \emptyset$
$cnt[\cdot] = 0$
for $r = 1$ to θ **do**
  $fl_r(s) \leftarrow$ Nodes infected by s under SICP model // Sample the influence of s
  for u_q in $fl_r(s)$ **do**
    $cnt[u_q] \leftarrow cnt[u_q] + 1$
  end for
  $fl(s) \leftarrow fl(s) \cup fl_r(s)$
end for
$\sigma(s) = \frac{1}{\theta} \sum_{r=1}^{\theta} |fl_r(s)|$ // Estimate the influence of s by average
$b = \lfloor \sigma(s) \rfloor$
$\widehat{fl}(s) \leftarrow$ Top b nodes in $fl(s)$ with the maximal $cnt[\cdot]$
$\mathcal{G}^{(k+1)} \leftarrow$ Remove s and it main followers $\widehat{fl}(s)$ from $\mathcal{G}^{(k)}$
return $\mathcal{G}^{(k+1)}$

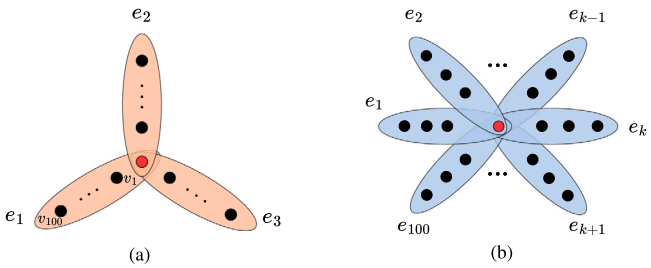


Fig. 5. Two different situation with the same degree 300.

caused when v_i infects its each susceptible neighbour in e_k with infection probability β. The hyperedge e_k chosen by the first step is crucial to the number of nodes which can be infected by v_i through e_k subsequently. Intuitively, more nodes e_k contains, more nodes might be infected by v_i through e_k.

An example is given in Fig. 5 to illustrate such intuition. Red colour refer to infected state and black colour refer to susceptible state. The red nodes in Fig. 5(a) and Fig. 5(b) both have the same degree 300, but their hyperdegree is different significantly. The hypergraph in Fig. 5(a) has only 3 hyperedges and each hyperedge contains 100 nodes (except the red node), while the hypergraph in Fig. 5(b) has 100 hyperedges and each hyperedge contains only 3 nodes (except the red node). In this scenario, The red nodes in Fig. 5(a) can infect 100 nodes at most within one time step no matter which hyperedge is chosen randomly in the first step. By contrast, the red nodes in Fig. 5(b) can infect only 3 nodes at most within one time step. The 1-hop expected infected node number of the red node in Fig. 5(a) is $100 * \beta$, while that of red node in Fig. 5(b) is only $3 * \beta$, under the infection probability β. These two red nodes have the same node degree but have very different 1-hop expected infected node number. It shows that it is unreasonable to select seed nodes only according to the maximal node degree.

We can conclude that node with more neighbours on each hyperedge tends to have larger influence. For node $v \in V$, we name the mean value of number of nodes each its connecting hyperedge contains as **Neighbourhood Coefficient**, denoted by $\alpha(v)$, where

$$\alpha(v) = \frac{1}{|E_v|} \sum_{e_i \in E_v} |e_i| = \frac{deg(v)}{Hdeg(v)}. \tag{18}$$

To validate the above observation, for each node $v_i \in V$, we calculate the neighbourhood coefficient $\alpha(v_i)$ and influence $\sigma(\{v_i\})$, and analyse whether there exists significant correlation between $\alpha(v_i)$ and $\sigma(\{v_i\})$. In practice, we estimate $\sigma(\{v_i\})$ by the average of R times of diffusion results, where R is set to 500. We set $\alpha(v_i)$ and $\sigma(\{v_i\})$ of each node v_i as coordinates respectively and plot them in the axes, as illustrated in Fig. 6. Besides, we use the Spearman's rank correlation coefficient (ρ) (Spearman, 1987) to conduct quantitative analysis for the correlation between these two statistics. The Spearman's rank correlation coefficient is a measure of the monotonicity of the relationship between two statistics. It varies from -1 to 1 with 0 implying no correlation, 1 implies exact positive monotonic relationship. And the more ρ approaches to 1, the stronger the positive monotonic relationship between the variables. We give the corresponding ρ over each dataset in Fig. 6. From Fig. 6, we can observe that nodes have larger neighbourhood coefficient α tends to have larger influence. Especially, on datasets Algebra, Restaurants-Rev, Geometry, Bars-Rev and iAF1260b, the node which has the maximal influence is the node with the maximal neighbourhood coefficient. Thus we can conclude that there exists significant

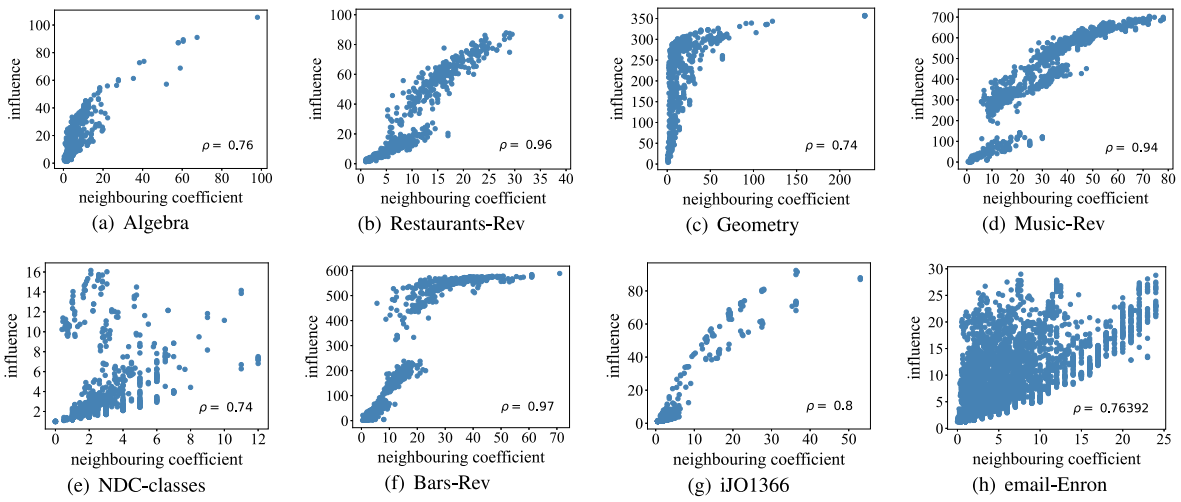


Fig. 6. Correlation analysis between nodes' influence and neighbourhood coefficient on 8 datasets.

positive monotonic relationship between the neighbourhood coefficient and influence. Therefore, we can develop a neighbourhood coefficient based method to solve the IM problem on hypergraph under the SICP model.

Algorithm 5 Adaptive Neighbourhood Coefficient (*Adeff*)

Input: Hypergraph $H(V, E)$, infection probability β, time step threshold T, seed set size K, number of simulations θ.
Output: Set of seed node S.

$H^{(0)}(V^{(0)}, E^{(0)}) \leftarrow H(V, E)$
$|S| = \emptyset$
while $|S| < K$ **do**
  $k \leftarrow |S|$
  $deg^{(k)} \leftarrow$ Degree of each node in $H^{(k)}(V^{(k)}, E^{(k)})$
  $Hdeg^{(k)} \leftarrow$ Hyperdegree of each node in $H^{(k)}(V^{(k)}, E^{(k)})$
  $\alpha^{(k)} \leftarrow \frac{deg^{(k)}}{Hdeg^{(k)}}$
  $s \leftarrow \arg\max_{v \in V^{(k)}} \alpha^{(k)}(v)$
  $S \leftarrow S \cup \{s\}$
  $cnt[\cdot] = 0$
  $fl(s) = \emptyset$
  for $r = 1$ to θ **do**
    $fl_r(s) \leftarrow$ Nodes infected by s in $H^{(k)}(V^{(k)}, E^{(k)})$ under diffusion model $\mathcal{M}(\beta, T)$
    for u_q in $fl_r(s)$ **do**
      $cnt[u_q] \leftarrow cnt[u_q] + 1$
    end for
    $fl(s) \leftarrow fl(s) \cup fl_r(s)$
  end for
  $\sigma(s) = \frac{1}{\theta} \sum_{r=1}^{\theta} |fl_r(s)|$ // Estimate the influence of s by average
  $b = \lfloor \sigma(s) \rfloor$
  $\widehat{fl}(s) \leftarrow$ Top b nodes in $fl(s)$ with the maximal $cnt[\cdot]$
  $H^{(k+1)}(V^{(k+1)}, E^{(k+1)}) \leftarrow$ Remove s and it main followers $\widehat{fl}(s)$ from $H^{(k)}(V^{(k)}, E^{(k)})$
end while
return S

4.2.2. Adaptive neighbourhood coefficient algorithm

Given a hypergraph $H(V, E)$, we can obtain the degree $deg(v_i)$ and hyperdegree $Hdeg(v_i)$ for each node v_i and computed the neighbourhood coefficient $\alpha(v_i)$ very quickly. It is intuitive to develop an approach which selects nodes with the top K maximal neighbourhood coefficient $\alpha(v_i)$ as the seed nodes. However, there would be a problem lying in such approach, *i.e.,* the influence overlap problem described in sub Section 4.1.6. To alleviate this problem, we select node with the maximal neighbourhood coefficient α adaptively by iteration. Specifically, in each iteration we select the node with the maximal neighbourhood coefficient

Table 2
Statistics of datasets. $|V|$ and $|E|$ represent the number of nodes and hyperedges respectively. $< deg >$ and $< Hdeg >$ represent the average degree and hyperdegree. $< |e| >$ represents the average number of nodes each hyperedge contains.

Datasets	$\|V\|$	$\|E\|$	$< deg >$	$< Hdeg >$	$< \|e\| >$
Algebra	423	1268	78.9	19.5	6.5
Restaurant-Rev	565	601	79.7	8.1	7.7
Geometry	580	1193	164.8	21.6	10.5
Music-Rev	1106	694	167.9	9.5	15.1
NDC-classes	1161	1088	10.7	5.6	5.9
Bars-Rev	1234	1194	174.3	9.6	9.9
iJO1366	1805	2546	16.9	5.6	5.9
email-Enron	4423	15,653	25.3	14.6	4.1

α and add it into the current seed node set, then we remove it and its main followers from the current hypergraph. α is recomputed in next iteration.

We name this method as **adaptive neighbourhood coefficient algorithm (*Adeff*)** and the details is presented in Algorithm 5:

- **Step 1:** At the beginning, with the input hypergraph H, we calculate the initial neighbourhood coefficient $\alpha^{(0)}$ by the ratio of degree to hyperdegree.
- **Step 2:** At step k, computing the adaptive neighbourhood coefficient $\alpha^{(k)}$ under current hypergraph $H^{(k)}$. Node v which has the maximal $\alpha^{(k)}$ is chosen as current seed node s and added to the seed node set S. Estimating the influence $\sigma(s)$ of node s by the average of θ times of diffusion sampling results, and selecting the nodes infected by s the most times during diffusion sampling as the main followers of s, denoted by $\widehat{fl}(s)$. Then removing s and its main followers $\widehat{fl}(s)$ from $H^{(k)}$, obtaining $H^{(k+1)}$.
- **Step 3:** The algorithm terminates when $|S| = K$.

5. Experiments

5.1. Experiment setup

Datasets. We use 8 real-world datasets, which are derived from different website[1][2] (Amburg, Kleinberg, & Benson, 2021; Benson, Abebe, Schaub, Jadbabaie, & Kleinberg, 2018). **Algebra & Geometry:** nodes represent users on MathOverflow website and a set of nodes will form a hyperedge if they answered the same type of question. **Restaurant-Rev:** nodes represent Yelp users and a set of nodes will form a hyperedge if they reviewed the same type of restaurant within a month. **Music-Rev:** nodes denote Amazon reviewers and a set of nodes will form a hyperedge if they reviewed the same type of blues music within a month. **NDC-classes:** nodes represent drug class labels, a set of labels will form a hyperedge if they consist the same drug. **iJO1366:** nodes are metabolics, and a set of nodes will form a hyperedge if they are applied to a certain reaction. **email-Enron:** nodes are email addresses, and a set of nodes will form a hyperedge if they are used in the same email simultaneously. Table 2 summarizes the statistics of these real-world hypergraphs, $< \cdot >$ denote the average.

Baseline. To verify the performance of our proposed methods (MIE and Adeff), we compare our methods with several baselines: (1) **HADP**. HADP algorithm is proposed in Xie, Zhan, Liu and Zhang (2023) and is the newest algorithm for this problem currently. It selects node with the maximal adaptive degree iteratively as the seed node set. (2) **HyperIMRANK** (MA & Rajkumar, 2022). A ranking based algorithms for the HyperCascade diffusion model (Gangal, Ravindran, & Narayanam, 2016) on hypergraphs with given influence diffusion probabilities *i.e.,* p_1 and p_2. It updates nodes' influence rank iteratively until the influence rank converges during iterations, then nodes in the top of rank are collected as the seed node set. In our experiments, we set p_1 and p_2 as 0.01 both, which is the same with that in Xie, Zhan, Liu and Zhang (2023) and collect the top K nodes in the convergent rank as the seed node set. (3) **Greedy**. Greedy selects one node with the maximal influence in each iteration until all seed nodes are selected in K iterations. In practice, to obtain the influence of a node we need to conduct many simulations and we set the times of simulations as 500. Due to the time complexity of the Greedy algorithm is extremely high, we constrain its running time within 48 hours in experiments. The details is presented in Algorithms 6. (4) **H-RIS**. Reverse influence sampling (RIS) (Borgs et al., 2014) algorithm is popular and has shown good performance in IM problem on ordinary graphs. Xie, Zhan, Liu and Zhang (2023) extended RIS from ordinary graph to hypergraph, H-RIS namely, and regard it as one of the baselines. In this work, we follow the same experimental configures in Xie, Zhan, Liu and Zhang (2023). (5) **Degree**. It selects top K nodes with the maximal node degree as the seed node set. (6) **HDegree**. It selects top K nodes with the maximal node hyperdegree as the seed node set.

[1] https://www.cs.cornell.edu/~arb/data/
[2] http://bigg.ucsd.edu/

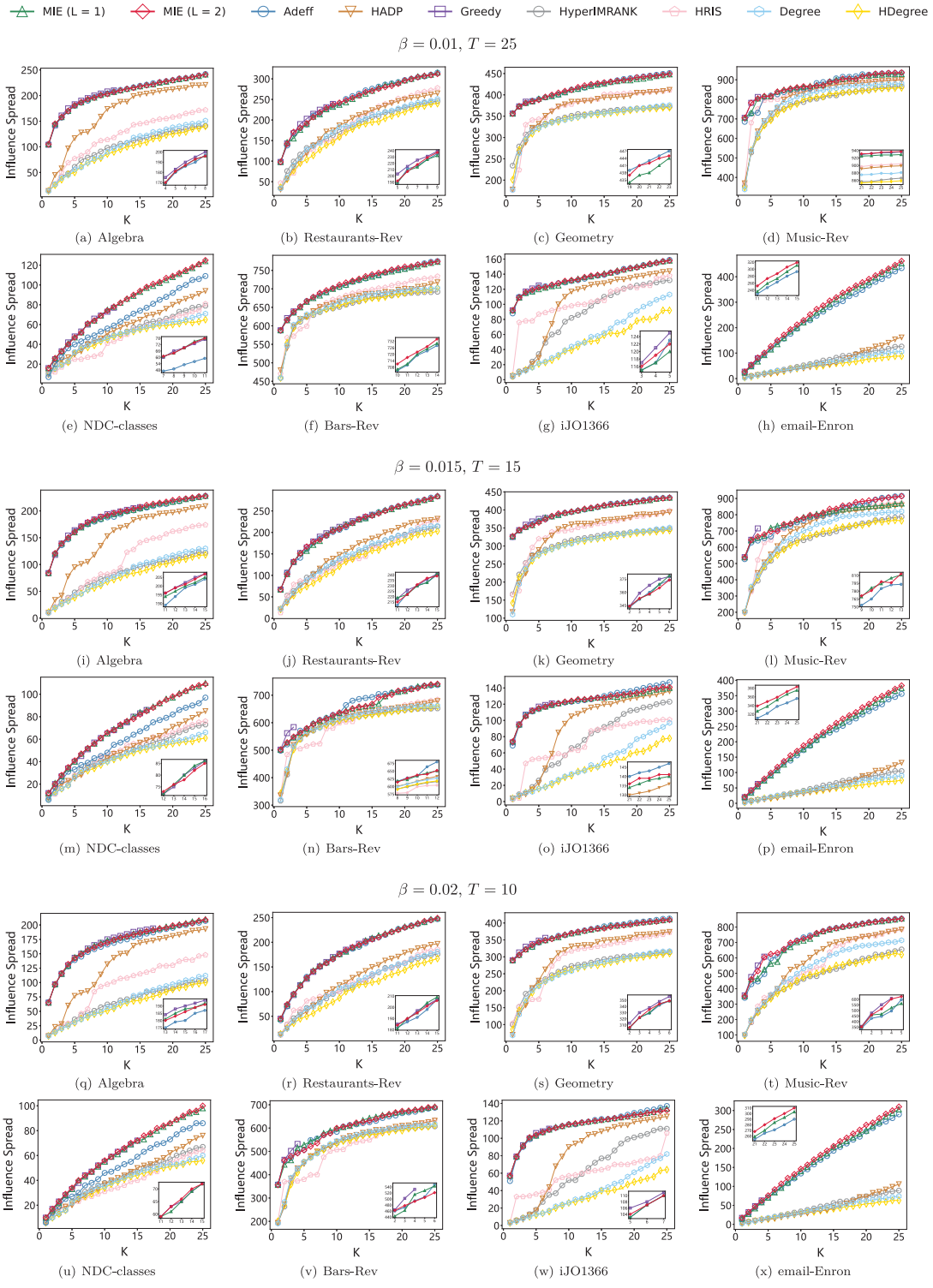


Fig. 7. Expected influence spread comparison by varying K for each algorithm on real-world datasets.

Algorithm 6 Greedy

Input: Hypergraph $H(V, E)$, infection probability β, time step threshold T, seed set size K, number of simulations θ.
Output: Set of seed node S.

$S = \emptyset$
while $|S| < K$ **do**
  $k \leftarrow |S|$
  for each node $v \in V$ **do**
    for $i = 1$ to θ **do**
      $\sigma_i(S \cup \{v\}) \leftarrow$ number of nodes infected by $S \cup \{v\}$ under diffusion model $\mathcal{M}(\beta, T)$
    end for
    $\sigma(S \cup \{v\}) = \frac{1}{\theta} \sum_{i=1}^{\theta} \sigma_i(S \cup \{v\})$ // Estimate the influence of v by average
  end for
  $s = \arg\max_{v \in V,\, v \notin S} \sigma(S \cup \{v\})$
  $S \leftarrow S \cup \{s\}$
end while
return S

Implementation. Following the previous works (Xie, Zhan, Liu & Zhang, 2023), we aim to find up to 25 seed nodes, *i.e.,* $K \leq 25$. We compare our methods, *i.e.,* MIE ($\mathsf{L} = 1$), MIE ($\mathsf{L} = 2$) and Adeff with baselines above in terms of effectiveness and efficiency. For infection probability β and time step threshold T, we set 3 different configures, *e.g.,* ($\beta = 0.01, T = 25$), ($\beta = 0.015, T = 15$) and ($\beta = 0.02, T = 10$) and conduct experiments respectively, which is the same with that in Xie, Zhan, Liu and Zhang (2023). All methods including our proposed methods and baselines are coded with Python and all experiments are conducted on a Linux server with single Intel(R) Xeon(R) Silver 4208 @2.10 GHz CPU.

5.2. Effectiveness on real-world datasets

To validate the effectiveness of our proposed approaches, we conduct extensive experiments on real-world hypergraph datasets. Specifically, we compare the influence (number of infected nodes) of seed node sets generated by our proposed approaches and baselines respectively, and we set various configuration for infection probability β and time step threshold T. For a seed node set, We conduct 500 times of influence diffusion under SICP model and we report the average of results. For the size of seed set, K, we vary it from 1 to 25. The overall experimental results are illustrated in Fig. 7. Specially, due to the high time complexity of the Greedy algorithm, it cannot work out all the 25 seed nodes within 48 hours, and we report the results of it within 48 hours running time. Note that HRIS do not finish running within 48 hours in email-Enron, thus we do not report its results for email-Enron dataset.

From Fig. 7, we have several observations. Firstly, our proposed approaches MIE ($\mathsf{L} = 1$), MIE ($\mathsf{L} = 2$) and *Adeff* outperform all baselines except Greedy on all datasets by a significant margin, and our proposed approaches have similar performance compare to Greedy. Secondly, MIE ($\mathsf{L} = 2$) has the best effectiveness comprehensively and can acquire about 450% improvement maximally compared to HADP, *e.g.,* in Fig. 7(h) when $K = 10$, and MIE ($\mathsf{L} = 1$) outperforms $Adeff$ in some cases, especially on datasets NDC-classes and email-Enron. Besides, MIE ($\mathsf{L} = 1$) and MIE ($\mathsf{L} = 2$) can always find the best seed node when $K = 1$ while $Adeff$ cannot in some cases, *e.g.,* in dataset NDC-classes. Another noticeable phenomenon is that MIE ($\mathsf{L} = 1$) and MIE ($\mathsf{L} = 2$) have the same effectiveness sometimes, *e.g.,* in Fig. 7(a) when K varies from 1 to 9, this is because in some scenarios only by 1-hop estimation MIE algorithm is sufficient to estimate the rank of influence expectation accurately, especially for the node with the maximal influence expectation, such that both MIE ($\mathsf{L} = 1$) and MIE ($\mathsf{L} = 2$) can find the node with the maximal influence expectation with respect to current seed node set and they output the same seed node set. Take an observation on results of baselines, intuitive methods like Degree and HDegree algorithm are very ineffective, which means that this problem cannot be solved by selecting seed nodes simply according to node degree and hyperdegree. Besides, extended method from ordinary graph IM problem like H-RIS and method applying on different diffusion model on hypergraphs like HyperIMRANK also failed to find high quality seed nodes. This shows that approaches applied on other domains cannot be used in this problem directly and also confirms the professionalism of our proposed methods.

In addition, we also compute the AUC (Area Under the Curve) value to comprehensively compare each method's effectiveness. We take $\beta = 0.02, T = 10$ and $K = 25$ as an example setup to compute the AUC metric. We report the AUC results and the performance boosts of our proposed methods w.r.t the best of baselines (except Greedy) in Table 3. From this table we can see that our proposed methods achieved significant performance boost (13.1% at least and 279.9% at most). On the other hand, each of our proposed methods are able to achieve SOTA effectiveness in different datasets. In general, MIE ($\mathsf{L} = 2$) has the best performance.

5.3. Ablation study

In this subsection, we investigate the effectiveness of our proposed influence overlap alleviation technique, which we both used in our proposed methods MIE and $Adeff$. We take $Adeff$ and MIE ($\mathsf{L} = 1$) as examples to compare them with their variants without influence overlap alleviation respectively with infection probability $\beta = 0.02$, $T = 10$. In the variants without influence overlap

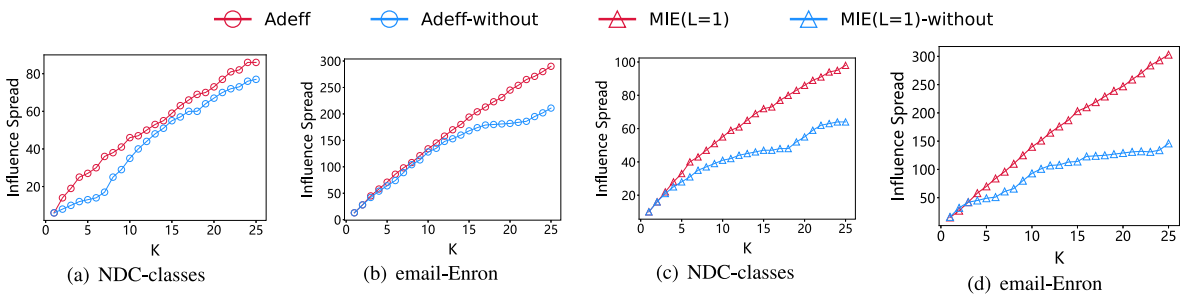


Fig. 8. Ablation study on influence overlap alleviation, $\beta = 0.02$, $T = 10$.

Table 3
AUC value computed by each curve from Fig. 7(q) to Fig. 7(x). Note that we have divided the original area value by 10,000 for clear presentation. $\beta = 0.02$, $T = 10$. The maximal AUC value among all methods is shown in bold and the maximal AUC value among baselines is shown with *.

Datasets	Our proposed methods			Baselines					Boost
	MIE (L = 1)	MIE (L = 2)	*Adeff*	HADP	HyperIMRANK	HRIS	Degree	Hdegree	
Algebra	**5.17**	5.13	5.04	4.00*	2.00	2.85	2.00	1.82	29.3% ↑
Restaurant-Rev	**5.51**	5.48	5.46	3.77*	3.43	3.59	3.46	3.07	46.2% ↑
Geometry	11.18	11.17	**11.22**	9.26*	8.01	8.93	7.81	7.74	21.2% ↑
Music-Rev	21.50	**21.91**	21.58	17.52*	14.34	16.83	16.18	14.34	25.0% ↑
NDC-classes	1.85	**1.88**	1.57	1.31*	1.25	1.13	1.17	1.13	43.5% ↑
Bars-Rev	**1.81**	1.80	1.79	1.60*	1.58	1.53	1.60	1.57	13.1% ↑
iJO1366	3.46	3.45	**3.47**	2.48*	1.89	1.74	1.12	0.97	39.9% ↑
email-Enron	5.05	**5.28**	4.92	1.31	1.39*	/	1.21	1.06	279.9% ↑

Table 4
Running time of each method. $\beta = 0.02$, $T = 10$ and $K = 15$. $48h+$ means the corresponding method did not finish running within 48 hours.

Datasets	Running time (seconds)								
	MIE (L = 1)	MIE (L = 2)	*Adeff*	HADP	Greedy	HyperIMRANK	HRIS	Degree	Hdegree
Algebra	25.93	99.01	15.04	7.91	83 044.94	8.35	217.30	0.9	0.1
Restaurant-Rev	28.40	192.88	14.03	4.26	85 315.26	26.52	88.92	0.68	0.13
Geometry	108.80	358.16	52.92	16.02	$48h+$	18.89	384.98	1.38	0.14
Music-Rev	138.78	702.16	39.41	11.88	$48h+$	104.27	605.43	1.69	0.27
NDC-classes	9.31	36.51	11.35	3.11	82 231.84	20.99	1518.79	1.23	0.28
Bars-Rev	225.92	1129.28	49.31	12.77	$48h+$	191.73	1282.21	1.82	0.28
iJO1366	38.61	179.60	34.67	8.46	$48h+$	65.62	8438.90	2.02	0.47
email-Enron	144.12	1260.12	104.41	36.26	$48h+$	442.99	$48h+$	10.46	1.00

alleviation we directly select nodes with the top K corresponding values as the seed nodes. Specifically, for $Adeff$, we compute the neighbourhood coefficient α for each node, then the top K nodes with the maximal α are selected as the seed node set. We name this variant as $Adeff$-without. Similarly, for MIE (L = 1), we compute the influence score (L = 1) for each node, then the top K nodes with the maximal influence score are selected as the seed node set. We name this variant as MIE (L =1)-without. Experimental results are shown in Fig. 8. We can observe that methods with influence overlap alleviation outperform their corresponding variants significantly both for $Adeff$ and MIE (L = 1), and the performance can be improved by up to 119%, *e.g.,* in Fig. 8(d) when $K = 24$. Therefore we can conclude that our proposed influence overlap alleviation technique is effective significantly.

5.4. Efficiency

In this subsection we investigate the efficiency of our proposed algorithms. We vary K from 5 to 15 and compare the time cost of outputting seed nodes for our proposed algorithms and HADP with $\beta = 0.02$, $T = 10$. We also test the running time of each method by setting $K = 15$. The experimental results are presented in Fig. 9 and Table 4. Among 3 algorithms this paper proposed, $Adeff$ has the best efficiency thus achieves the best trade-off between effectiveness and efficiency. By compare the time cost of MIE (L = 1) and MIE (L = 2), we can find that the time cost of MIE algorithm increase sharply with the increase of search path length constraint L. This phenomenon is intuitive since the number of infection path increases exponentially as L grows so that the time cost for searching all the infection paths also grows exponentially. However, in later subsection, we analysed L = 1 and L = 2 is sufficient for most cases to estimate the influence expectation rank and there is no need to use deeper path length, thus avoiding unacceptable time consuming of this algorithm. Finally, we can conclude that the 3 algorithms this paper proposed have different

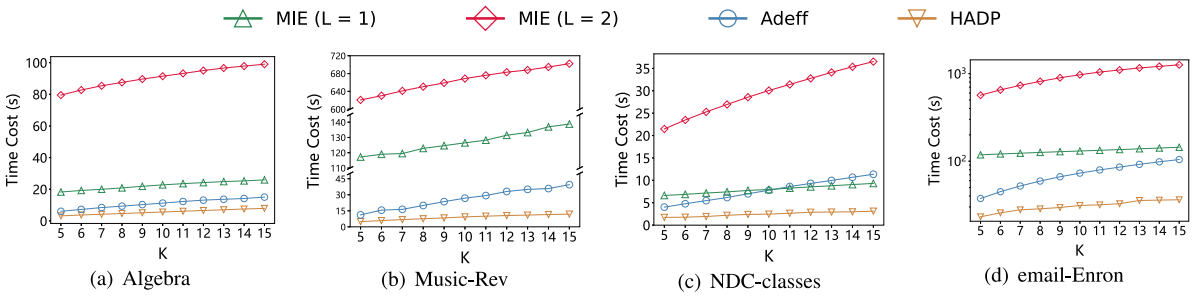


Fig. 9. Comparison on efficiency, $\beta = 0.02$, $T = 10$.

Table 5
Influence spreads comparison when varying parameters β or T respectively. $K = 25$. Fixing $T = 15$, varying β from $\{0.01, 0.02, 0.03, 0.04, 0.05\}$ and fixing $\beta = 0.03$, varying T from $\{5, 10, 15, 20, 25\}$ in MIE ($\mathrm{L} = 2$). The influence spreads of all seed sets are tested under $\beta = 0.03, T = 15$, and results is averaged by 500 times of spread simulations.

Datasets	β					T				
	0.01	0.02	0.03	0.04	0.05	5	10	15	20	25
Restaurant-Rev	389	422	423	410	377	386	424	423	412	374
NDC-classes	168	201	201	198	187	167	202	201	197	188
email-Enron	911	942	918	901	858	899	936	918	896	851

Table 6
Mean effectiveness score over K seed nodes detected by MIE ($\mathrm{L} = 1$) and MIE ($\mathrm{L} = 2$). The effectiveness score of seed node s_k in $\mathcal{G}^{(k)}$ is computed by $\frac{\sigma(s_k)}{\sigma(s_k^*)}$, where s_k^* denotes the best seed node in $\mathcal{G}^{(k)}$. $K = 25$, $\beta = 0.02$, $T = 10$.

Mean effectiveness score ↑								
	Algebra	Restaurant-Rev	Geometry	Music-Rev	NDC-classes	Bars-Rev	iJO1366	email-Enron
MIE ($\mathrm{L} = 1$)	0.983	0.987	0.950	0.871	0.966	0.967	0.982	0.884
MIE ($\mathrm{L} = 2$)	0.992	0.988	0.980	0.964	0.976	0.986	0.989	0.952

effectiveness level and efficiency level, and algorithm with better effectiveness also have lower efficiency inevitably, $Adeff$ and MIE ($\mathrm{L} = 1$) achieve good trade-off between effectiveness and efficiency.

5.5. Parameters sensitivities

In this subsection we investigate the parameters sensitivities of MIE about infection probability β and time step threshold T. In practice, we set $\beta = 0.03, T = 15$ as the uniform setup when evaluating the influence spread of seed nodes. We investigate the sensitivity in terms of β and T respectively: fixing $T = 15$, varying β from {0.01, 0.02, 0.03, 0.04, 0.05} and fixing $\beta = 0.03$, varying T from {5, 10, 15, 20, 25}. The number of seed nodes K is set as 25. We feed different setups of β and T mentioned above in MIE ($\mathrm{L} = 2$) algorithm to obtain different seed sets, and then test their influence spreads results under the same setup, *i.e.,* $\beta = 0.03, T = 15$. The comparison results on 3 datasets are reported in Table 5. From Table 5 we can find that when β or T deviate from the test value (0.03 for β, 15 for T) to a large extent, the influence spread will also fluctuate greatly. For example, in Restaurant-Rev dataset, when β changed from 0.03 to 0.05, the influence spread dropped from 423 to 377. In addition, generally the influence spread is less sensitive relatively when β or T changes slightly. For example, in NDC-classes dataset, when T changed from 15 to 20, the influence spread changed only from 201 to 197. A surprising result is that, the influence spread increased from 918 to 942 when β changed to 0.02 in email-Enron dataset and similar phenomenon can also be observed when varying T. By further analysis in email-Enron we find that the influence spread of the first seed node selected by MIE ($\mathrm{L} = 2, \beta = 0.02$) is only 70, which is smaller than that of MIE ($\mathrm{L} = 2, \beta = 0.03$), *i.e.,* 83. However, MIE ($\mathrm{L} = 2, \beta = 0.02$) can outperform MIE ($\mathrm{L} = 2, \beta = 0.03$) gradually with the growth of K. This phenomenon shows the common drawback of greedy frameworks: greedy algorithms will easily fall into local optimum thus fail to reach global optimum, and also shows solving the HyperIM problem well is challenging.

5.6. Rank correlation analysis

In this subsection we investigate to what extent multi-hop influence score can accurately estimate the rank of influence expectation. Similar with Section 4.2, we use Spearman's rank correlation coefficient (Spearman, 1987), denoted by ρ, to conduct quantitative analysis for the correlation between the L-hop influence score computed by MIE ($\mathrm{L} = 1, 2, 3$) and influence expectation

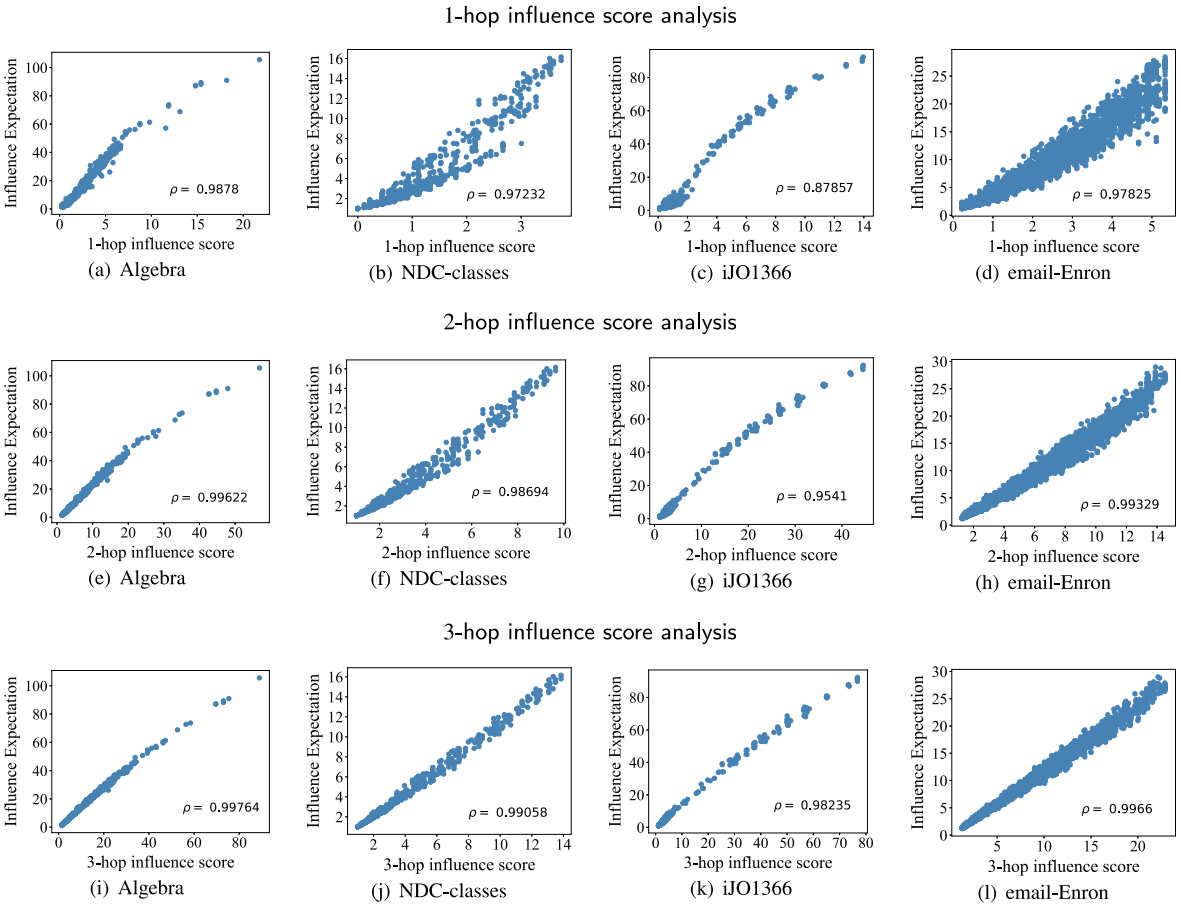
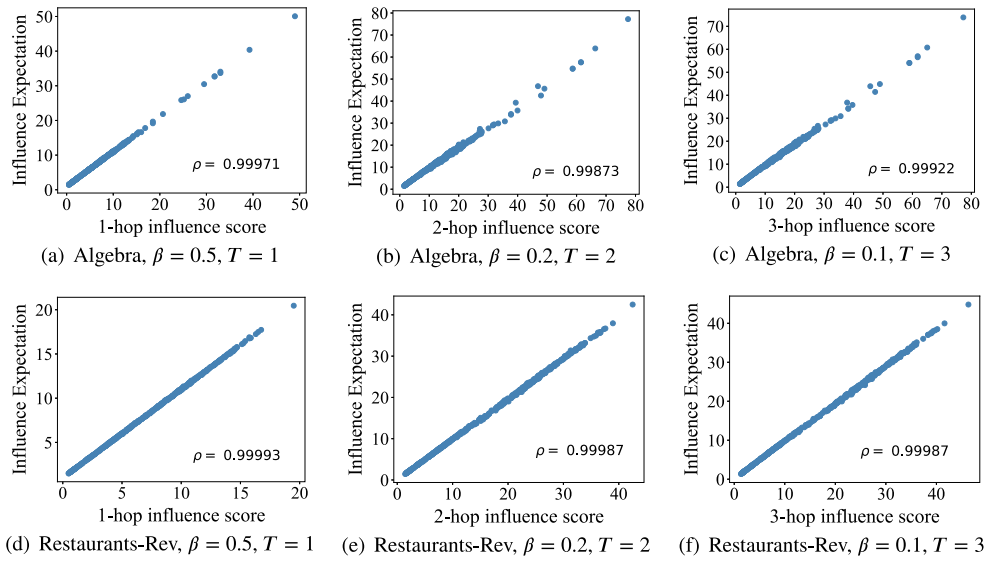


Fig. 10. Correlation analysis between nodes' influence expectation and multi-hop influence score.

on 4 datasets, $\beta = 0.01$, $T = 25$. The influence expectation is computed by the average of 500 times of diffusion results. We visualize the experimental results and illustrate them in Fig. 10. We have several observations. Firstly, it shows significant positive monotonic relationship between influence expectation and L-hop influence score. Secondly, for the same dataset deeper path length constraint L brings higher correlation coefficient and ρ can reach 0.99, which is very close to 1., *e.g.,* ρ grows from about 0.879 to 0.982 with the path length constraint L grows from 1 to 3 in iJO1366. Besides, we can see that node with the maximal L-hop influence score also has the maximal influence expectation even when $\mathsf{L} = 1$ or $\mathsf{L} = 2$ in dataset Algebra, NDC-classes and JO1366. Thus we can say that $\mathsf{L} = 1$ and $\mathsf{L} = 2$ is sufficient to estimate the rank of influence expectation in most cases.

Further, we also conduct experiments to investigate to what extent multi-hop influence score can accurately estimate the rank of influence expectation when $\mathsf{L} = T$. We set three different combinations of β and T: $(\beta = 0.5, T = 1)$, $(\beta = 0.2, T = 2)$ and $(\beta = 0.1, T = 3)$. For each combination, we set $\mathsf{L} = T$ and we compute the influence expectation of each node by the average of 10,000 times of diffusion results. We illustrate the experimental results in Fig. 11. We can see the correlation coefficient ρ is very close to 1, *e.g.,* $\rho = 0.99993$ in Fig. 11(d). This supports what we claim in this paper earlier, *i.e.,* for node v_i, if $\mathsf{L} = T$, $\Delta[\mathsf{v}_i] = \mathbb{E}(\sigma_{\mathcal{G}}(\mathsf{v}_i))$.

Furthermore, we conduct more detailed experiments to verify the choice of 1-hop and 2-hop estimation in MIE is effective and sufficient. In MIE algorithm, the input hypergraph H is firstly transformed to a directed weighted graph $\mathcal{G}^{(0)}$. Then MIE computes influence score $\Delta[\cdot]$ and updates current graph $\mathcal{G}^{(k)}$ iteratively until K seed nodes are obtained during iterations. To verify the effectiveness of each seed node s_k obtained by MIE ($\mathsf{L} = 1$ or 2) in each iteration graph $\mathcal{G}^{(k)}$, we further find the best seed node s_k^* (node with the maximal influence spread) in $\mathcal{G}^{(k)}$ and compare their influence spread in $\mathcal{G}^{(k)}$, *i.e.,* $\sigma(s_k)$ and $\sigma(s_k^*)$. We set $K = 25$, thus we have K seed nodes obtained by MIE ($\mathsf{L} = 1$ or 2), $S = \{s_1, s_2, \ldots, s_K\}$, and K best seed nodes $S^* = \{s_1^*, s_2^*, \ldots, s_K^*\}$. We evaluate the effectiveness score of each seed node s_k by $\frac{\sigma(s_k)}{\sigma(s_k^*)}$. The mean values over K effectiveness scores, $\frac{1}{K}\sum_{k=1}^{K}\frac{\sigma(s_k)}{\sigma(s_k^*)}$, on 8 datasets are shown in Table 6. We can find that MIE ($\mathsf{L} = 1$) can reach 0.95 in most datasets and MIE ($\mathsf{L} = 2$) can achieve that in all datasets. Besides, MIE ($\mathsf{L} = 2$) are able to achieve up to 10.7% improvement by comparing with MIE ($\mathsf{L} = 1$), *e.g.,* in Music-Rev. Such results justified the choices of 1-hop and 2-hop estimation in MIE.



(a) Algebra, $\beta = 0.5, T = 1$ (b) Algebra, $\beta = 0.2, T = 2$ (c) Algebra, $\beta = 0.1, T = 3$

(d) Restaurants-Rev, $\beta = 0.5, T = 1$ (e) Restaurants-Rev, $\beta = 0.2, T = 2$ (f) Restaurants-Rev, $\beta = 0.1, T = 3$

Fig. 11. Correlation analysis between influence expectation and multi-hop influence score when $\mathsf{L} = T$.

5.7. Discussion of ethics

This paper focuses on maximizing the influence or information spread in social networks, *e.g.,* advertisement putting and public opinion propagation. Some ethical problems probably come with that are privacy disclosure and rumour spread. Note that our proposed algorithms (MIE and $Adeff$) achieve influence maximization only by utilizing graph structural information and are agnostic to users' privacy like gender and occupation. Thus, there are no privacy issues with our proposed algorithms. On the other hand, we can mitigate the risks of rumour spread by blocking the seed nodes detected by our proposed methods, if we find our proposed methods are misused. There are positive and negative information simultaneously in social networks, while information spread problem can also be categorized into influence maximization and influence minimization. Influence maximization studies like this paper aims to enhance positive information spread, while influence minimization studies aims to prevent negative information (e.g., rumour) spread. Influence minimization achieves its goal mainly by blocking a part of nodes or edges in a network, and in recent years there are many excellent works studying this problem (Manouchehri, Helfroush, & Danyali, 2021; Medya, Silva, & Singh, 2020; Teng, Xie, Zhang, Wang, & Zhang, 2023; Xie, Zhang, Wang, Lin, Zhang & Wenjie, 2023; Yang, Li, & Giua, 2019; Zareie & Sakellariou, 2021). Specially, existing influence minimization algorithms are beneficial and helpful when influence maximization algorithms like MIE and $Adeff$ are misused for rumour spread.

6. Conclusion

Influence maximization on hypergraphs is of great research significance for social network marketing. In this paper, we study this problem under the SICP model. By theoretical analysis we transform the hypergraph under SICP model to a directed weighted ordinary graph under a SICP-variant model. Further we proposed a method applied on the transformed graph under SICP-variant model, *i.e.,* MIE. Specifically, MIE estimates the rank of influence expectation for all nodes via computing influence scores by means of probability distribution model. We analysed the existence of influence expectation for a node, and we provided the method of computing influence expectation exactly. In addition, We proposed an simple but effective degree and hyperdegree based approach in terms of the characteristics of the SICP diffusion model, *i.e., Adeff*. Extensive experiments on 8 real-world datasets verify that our proposed MIE and *Adeff* algorithms significantly outperform the newest method.

CRediT authorship contribution statement

Xulu Gong: Conceptualization, Methodology, Validation, Writing – original draft, Software. **Hanchen Wang:** Conceptualization, Writing – review & editing, Supervision. **Xiaoyang Wang:** Methodology, Writing – review & editing. **Chen Chen:** Conceptualization, Writing – review & editing. **Wenjie Zhang:** Data curation, Writing – review & editing. **Ying Zhang:** Funding acquisition, Supervision.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Zhejiang Provincial Natural Science Foundation (ZJNSF) [No. LY21F020012] and Australian Research Council (ARC) [No. FT210100303].

References

Aghaee, Z., Ghasemi, M. M., Beni, H. A., Bouyer, A., & Fatemi, A. (2021). A survey on meta-heuristic algorithms for the influence maximization problem in the social networks. *Computing*, *103*, 2437–2477.

Aktas, M. E., Jawaid, S., Gokalp, I., & Akbas, E. (2022). Influence maximization on hypergraphs via similarity-based diffusion. In *2022 IEEE international conference on data mining workshops* (pp. 1197–1206). IEEE.

Amato, F., Moscato, V., Picariello, A., & Sperlí, G. (2017). Influence maximization in social media networks using hypergraphs. In *Green, pervasive, and cloud computing: 12th international conference, GPC 2017, Cetara, Italy, May 11-14, 2017, proceedings. Vol. 12* (pp. 207–221). Springer.

Amburg, I., Kleinberg, J., & Benson, A. R. (2021). Planted hitting set recovery in hypergraphs. *Journal of Physics: Complexity*, *2*(3), Article 035004.

Antelmi, A., Cordasco, G., Spagnuolo, C., & Szufel, P. (2021). Social influence maximization in hypergraphs. *Entropy*, *23*(7), 796.

Banerjee, S., Jenamani, M., & Pratihar, D. K. (2020). A survey on influence maximization in a social network. *Knowledge and Information Systems*, *62*, 3417–3455.

Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A., & Kleinberg, J. (2018). Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, *115*(48), E11221–E11230.

Borgs, C., Brautbar, M., Chayes, J., & Lucier, B. (2014). Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms* (pp. 946–957). SIAM.

Cai, T., Li, J., Mian, A., Li, R.-H., Sellis, T., & Yu, J. X. (2020). Target-aware holistic influence maximization in spatial social networks. *IEEE Transactions on Knowledge and Data Engineering*, *34*(4), 1993–2007.

Chen, S., Fan, J., Li, G., Feng, J., Tan, K.-l., & Tang, J. (2015). Online topic-aware influence maximization. *Proceedings of the VLDB Endowment*, *8*(6), 666–677.

Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 199–208).

Cheng, S., Shen, H., Huang, J., Chen, W., & Cheng, X. (2014). Imrank: influence maximization via finding self-consistent ranking. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 475–484).

Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., & Wu, Y. (2010). Graph pattern matching: From intractable to polynomial time. *Proceedings of the VLDB Endowment*, *3*(1–2), 264–275.

Gangal, V., Ravindran, B., & Narayanam, R. (2016). HEMI: Hyperedge majority influence maximization. In M. G. Armentano, A. Monteserin, J. Tang, & V. Yannibelli (Eds.), *CEUR workshop proceedings: vol. 1622, Proceedings of the 2nd international workshop on social influence analysis co-located with 25th international joint conference on artificial intelligence (IJCAI 2016), New York city, New York, USA, July 9, 2016* (pp. 38–47). CEUR-WS.org, URL: http://ceur-ws.org/Vol-1622/SocInf2016_Paper4.pdf.

Gomez-Rodriguez, M., Song, L., Du, N., Zha, H., & Schölkopf, B. (2016). Influence estimation and maximization in continuous-time diffusion networks. *ACM Transactions on Information Systems (TOIS)*, *34*(2), 1–33.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 137–146).

Kumar, S., Mallik, A., Khetarpal, A., & Panda, B. (2022). Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences*, *607*, 1617–1636.

Li, H., Bhowmick, S. S., Cui, J., Gao, Y., & Ma, J. (2015). Getreal: Towards realistic selection of influence maximization strategies in competitive networks. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* (pp. 1525–1537).

Li, J., Cai, T., Deng, K., Wang, X., Sellis, T., & Xia, F. (2020). Community-diversified influence maximization in social networks. *Information Systems*, *92*, Article 101522.

Li, Y., Fan, J., Wang, Y., & Tan, K.-L. (2018). Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *30*(10), 1852–1872.

Li, R.-H., Qin, L., Ye, F., Wang, G., Yu, J. X., Xiao, X., et al. (2020). Finding skyline communities in multi-valued networks. *The VLDB Journal*, *29*, 1407–1432.

Ma, C., Fang, Y., Cheng, R., Lakshmanan, L. V., Zhang, W., & Lin, X. (2020). Efficient algorithms for densest subgraph discovery on large directed graphs. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data* (pp. 1051–1066).

MA, A., & Rajkumar, A. (2022). Hyper-IMRANK: Ranking-based influence maximization for hypergraphs. In *5th Joint international conference on data science & management of data* (pp. 100–104).

Manouchehri, M. A., Helfroush, M. S., & Danyali, H. (2021). A theoretically guaranteed approach to efficiently block the influence of misinformation in social networks. *IEEE Transactions on Computational Social Systems*, *8*(3), 716–727.

Medya, S., Silva, A., & Singh, A. (2020). Approximate algorithms for data-driven influence limitation. *IEEE Transactions on Knowledge and Data Engineering*, *34*(6), 2641–2652.

Morone, F., & Makse, H. A. (2015). Influence maximization in complex networks through optimal percolation. *Nature*, *524*(7563), 65–68.

Singh, S. S., Srivastva, D., Verma, M., & Singh, J. (2022). Influence maximization frameworks, performance, challenges and directions on social network: A theoretical study. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 7570–7603.

Spearman, C. (1987). The proof and measurement of association between two things. *The American journal of psychology*, *100*(3/4), 441–471.

Su, X., & Zhang, Z. (2023). Causal influence maximization in hypergraph. arXiv preprint arXiv:2301.12226.

Tang, Y., Shi, Y., & Xiao, X. (2015). Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* (pp. 1539–1554).

Tang, Y., Xiao, X., & Shi, Y. (2014). Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on management of data* (pp. 75–86).

Teng, S., Xie, J., Zhang, M., Wang, K., & Zhang, F. (2023). IMinimize: A system for negative influence minimization via vertex blocking. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 5101–5105).

Wang, X., Zhang, Y., Zhang, W., & Lin, X. (2016a). Distance-aware influence maximization in geo-social network. In *ICDE* (pp. 1–12).

Wang, X., Zhang, Y., Zhang, W., & Lin, X. (2016b). Efficient distance-aware influence maximization in geo-social networks. *IEEE Transactions on Knowledge and Data Engineering*, *29*(3), 599–612.

Wang, X., Zhang, Y., Zhang, W., Lin, X., & Chen, C. (2016). Bring order into the samples: A novel scalable method for influence maximization. *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 243–256.

Xie, M., Zhan, X.-X., Liu, C., & Zhang, Z.-K. (2023). An efficient adaptive degree-based heuristic algorithm for influence maximization in hypergraphs. *Information Processing & Management*, *60*(2), Article 103161.

Xie, J., Zhang, F., Wang, K., Lin, X., & Zhang, W. (2023). Minimizing the influence of misinformation via vertex blocking. In *2023 IEEE 39th international conference on data engineering* (pp. 1–12). IEEE.

Yan, Q., Huang, H., Gao, Y., Lu, W., & He, Q. (2017). Group-level influence maximization with budget constraint. In *Database systems for advanced applications: 22nd international conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, proceedings, Part I. Vol. 22* (pp. 625–641). Springer.

Yang, L., Li, Z., & Giua, A. (2019). Influence minimization in linear threshold networks. *Automatica*, *100*, 10–16.

Zareie, A., & Sakellariou, R. (2021). Minimizing the spread of misinformation in online social networks: A survey. *Journal of Network and Computer Applications*, *186*, Article 103094.

Zhu, J., Zhu, J., Ghosh, S., Wu, W., & Yuan, J. (2018). Social influence maximization in hypergraph in social networks. *IEEE Transactions on Network Science and Engineering*, *6*(4), 801–811.