Computational Social Networks

RESEARCH Open Access

Influence maximization in social media networks concerning dynamic user behaviors via reinforcement learning

Mengnan Chen[1], Qipeng P. Zheng[1*], Vladimir Boginski[1] and Eduardo L. Pasiliao[2]

*Correspondence:
qipeng.zheng@ucf.edu
[1] Department of Industrial Engineering and Management Systems, University of Central Florida, 12800 Pegasus Drive, Orlando, FL, USA
Full list of author information is available at the end of the article

Abstract

This study examines the influence maximization (IM) problem via information cascades within random graphs, the topology of which dynamically changes due to the uncertainty of user behavior. This study leverages the discrete choice model (DCM) to calculate the probabilities of the existence of the directed arc between any two nodes. In this IM problem, the DCM provides a good description and prediction of user behavior in terms of following or not following a neighboring user. To find the maximal influence at the end of a finite-time horizon, this study models the IM problem by using multi-stage stochastic programming, which can help a decision-maker to select the optimal seed nodes by which to broadcast messages efficiently. Since computational complexity grows exponentially with network size and time horizon, the original model is not solvable within a reasonable time. This study then uses two different approaches by which to approximate the optimal decision: myopic two-stage stochastic programming and reinforcement learning via the Markov decision process. Computational experiments show that the reinforcement learning method outperforms the myopic two-stage stochastic programming method.

Keywords: Social networks, Markov decision process, Discrete choice model, Influence maximization, Multistage stochastic programming, Reinforcement learning

Introduction

Information maximization has played a vital role in various areas in human history, including politics, marketing, and both cultural and military campaigns. One common way of maximizing influence is via information cascades among the targeting population. Today, the rise of social media networks (e.g., Facebook, Twitter, and Snapchat) has greatly facilitated information dissemination in mass society; the literature has thoroughly and mathematically addressed this cascading spread of information. In mathematical terms, a network (including nodes/vertices, arcs/links/edges, and their states) is used to describe the dynamics of a real-world social media network. When a node of a network adopts certain information, it is "activated" [1]. As the definition presented in [2] states, an *activation sequence* is an ordered set of nodes that captures the order in which network nodes will adopt a piece of information. The first node in the activation

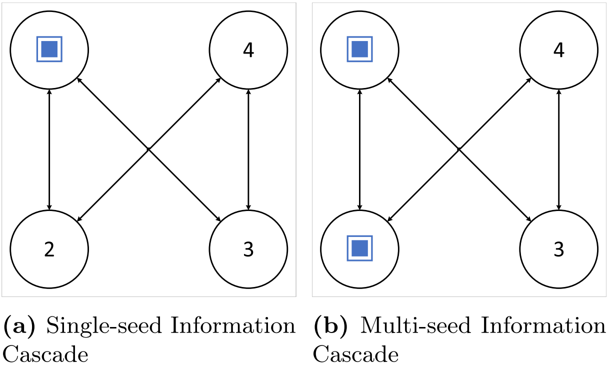
SpringerOpen

sequence is the *seed node*, and a *spreading cascade* is a directed tree that has as its root that first node. The tree captures the influence between nodes (with branches that represent who transmitted the information, and to whom), and it unfolds in the same order as the activation sequence. There are two typical information diffusion models—namely, independent cascade [3] and linear threshold [4]. The differences between these two models are as follows:

- Independent cascade: if x_{t-1} is the set of newly activated nodes or we call it seed node at time step $t-1$, then, at each time step t, each node i belonging to x_{t-1} will infect the inactive neighbor j with the probability p_{ij}.
- Linear threshold: if each node i has a threshold θ_i in the interval [0, 1], then, at each time step t, each inactive node j becomes active if the total inference from all activated neighbors $\sum_{i \in H_{t-1}} b_{ij} > \theta_j$, where H_{t-1} is the set of nodes activated at time $t-1$ or earlier.

Our study is based on the assumption of the independent cascade model. Among many pioneering studies, the authors of [5] propose the expectation maximization algorithm to predict information diffusion probabilities in the independent cascade model. The authors of [6] apply the influence maximization problem with an independent cascade model in prevailing viral marketing. Furthermore, the authors of [7] showed for the first time that the computing influence spread in the independent cascade model is NP-Hard; these studies have led to the design of a new heuristic algorithm that can easily scale up, relative to the greedy algorithm proposed in [1]. The influence maximization problem involves finding the nodes for the initial injection of information so as to maximize influence in a given social network with diffusion probabilities.

Our problem is a special case of the influence maximization problem. Our independent cascade model is based on the assumption that there are multiple seeds that broadcast information to the whole network. Figure 1 shows the difference between the multi-seed independent cascade model and a traditional single-seed model. In the single-seed model, node 1 is selected as the seed, and the minimal time needed to broadcast to the whole network is 2. In the multi-seed model, however, both nodes 1 and 2 are selected as seeds, and each node can receive the information within one



(a) Single-seed Information Cascade (b) Multi-seed Information Cascade

Fig. 1 Information cascade based on seed type

time period. If every node in the network is a seed, the minimal broadcast time will be 0, but this is not an economical method (owing to the seed cost). We look to pinpoint an efficient means of seed selection that can balance seed cost and broadcast time.

Unlike previous research on the independent cascade model, we assume that information diffusion probabilities or network topology probabilities dynamically change according to user behavior. Among the studies of dynamic user behavior, [8] introduces the concept of behavior change support systems. Based on that work, the authors of [9] found ample evidence of the strong influence exerted by social interaction on people's behaviors. The authors of [10] conducted extensive statistical analysis on large-scale real data and found that the general form of exponential, Rayleigh, and Weibull distribution can effectively preserve the characteristics of behavioral dynamics. The networked Weibull regression model for behavioral dynamics modeling is found to significantly improve the interpretability and generality of traditional survival models in [10].

Cascading phenomena are typically characterized by a dynamic process of information propagation among the nodes of a network, where nodes can repost information after seeing it posted by their neighbors. Moreover, the content and value of information may affect not only the reach (or depth) of a cascade, but also the topology of the underlying network; this is due to effects whereby nodes may either sever their ties with neighboring nodes (where the transmitted information is deemed unreliable, malicious, or both) or form new ties with nodes that transmit "reliable" information. In an independent cascade, people observe the choices of others and make decisions based on these observations, while concurrently considering their own personal preferences.

This phenomenon arises frequently in the field of behavioral economics and other social sciences. One real-world example is viral marketing, in which an independent cascade spreads information about a product with other people in their social networks, with the objective of promoting a product by leveraging existing social networks. A recent study of social networks [11] suggests that such processes may occur in a "bursty" fashion-that is, the patterns of network links change abruptly as a result of significant independent cascades. Thus, new information may create within a network a "burst" of node activations and edge activations/deactivations. In a decentralized autonomous network, agents or nodes act independently and behave according to their utility functions. To model their autonomous behaviors, we implement the concepts of discrete choice models, as drawn from behavioral economics [12, 13].

Our contributions Bearing in mind the endpoint of maximizing the influence of the information provider within a limited time, we model our problem as a seed selection problem of information spreading in dynamic networks that feature a random topology. In a social network, each user can have as many as three roles—namely, source user, message sender (i.e., followee of neighbors), and message receiver (i.e., follower of neighbors). It is possible that one node can play these three different roles at different times. For example, Alice writes and posts a message on a social media network. At that moment, she is the source user. In this network, Alice and Bob are friends, which means Alice is following Bob and Bob is also following Alice. Bob sees Alice's message, and so Bob is a receiver of Alice's message. Bob likes this message and reposts it to his own followers. At this point Bob is also message sender, given his "repost" action. Since Alice is also Bob's follower, she sees that her message is reposted, and at this point she becomes a message receiver with respect to the reposted message.

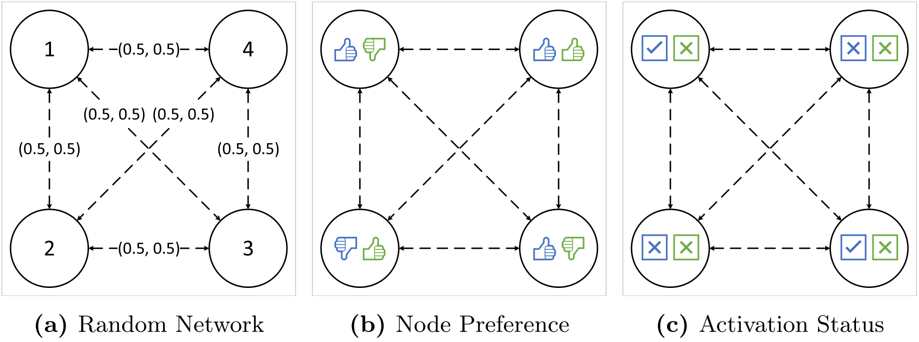
Generally, we can decompose our problem into two steps, as follows:

- Seed selection: this can be controlled by the information provider, who selects a proper set of initial seeds that will receive the deemed information. In the previous example, Alice is the seed node.
- Information cascade: this includes two variables. One is the node activation status, which describes the process wherein the user receives a message from their followee. The other one is the node repost decision, which is controlled by the message receiver. In our model, the repost decision depends on the user preference and the topic of the received message. In the previous example, Bob reposted the message because he likes it. However, if Bob dislikes this message, what will be happen? Since the message is coming from Alice, Bob may think Alice has tastes different from his, and so he might unfollow Alice. The "unfollow" action will break the information flow from Alice to Bob, which leads to a change in the network topology.

In this study, we propose an information maximization model through independent cascades, with random graphs. The network size and node preference is assumed to be given, while the friendship between any two users (i.e., arc connection) dynamically changes. Our model can help decision-makers choose the optimal action when they face an uncertain network topology. The stochastic formulation considers endogenous uncertainty, which is represented by the binary choice probability distribution of arc connection between any two nodes. To solve this problem, we design two problem-specific algorithms: one involves two-stage stochastic programming with a myopic policy, while the other involves reinforcement learning and the Markov decision process. We summarize the contributions of this study as follows:

- We introduce the discrete choice model in the information maximization problem, where the network topology dynamically changes during the independent cascading process.
- We develop practical algorithms to solve the multistage stochastic programming problem under endogenous uncertainty.
- To avoid directly dealing with large state spaces of node activation, we exploit the implicit Monte Carlo-based partially observable Markov decision process.
- We compare the results using two algorithms and various sample sizes.

The remainder of this paper is structured as follows. After having briefly described information maximization and the independent cascade problem in random graphs within a finite-time horizon, we provide in "Mathematical models" section the original multistage stochastic programming models with several assumptions. In "Solution approaches" section, we design two algorithms to solve this problem. The computational results are presented in "Computational experiments on algorithms' convergence" section, while "Conclusion" section provides concluding remarks.



(a) Random Network **(b)** Node Preference **(c)** Activation Status

Fig. 2 Given network properties

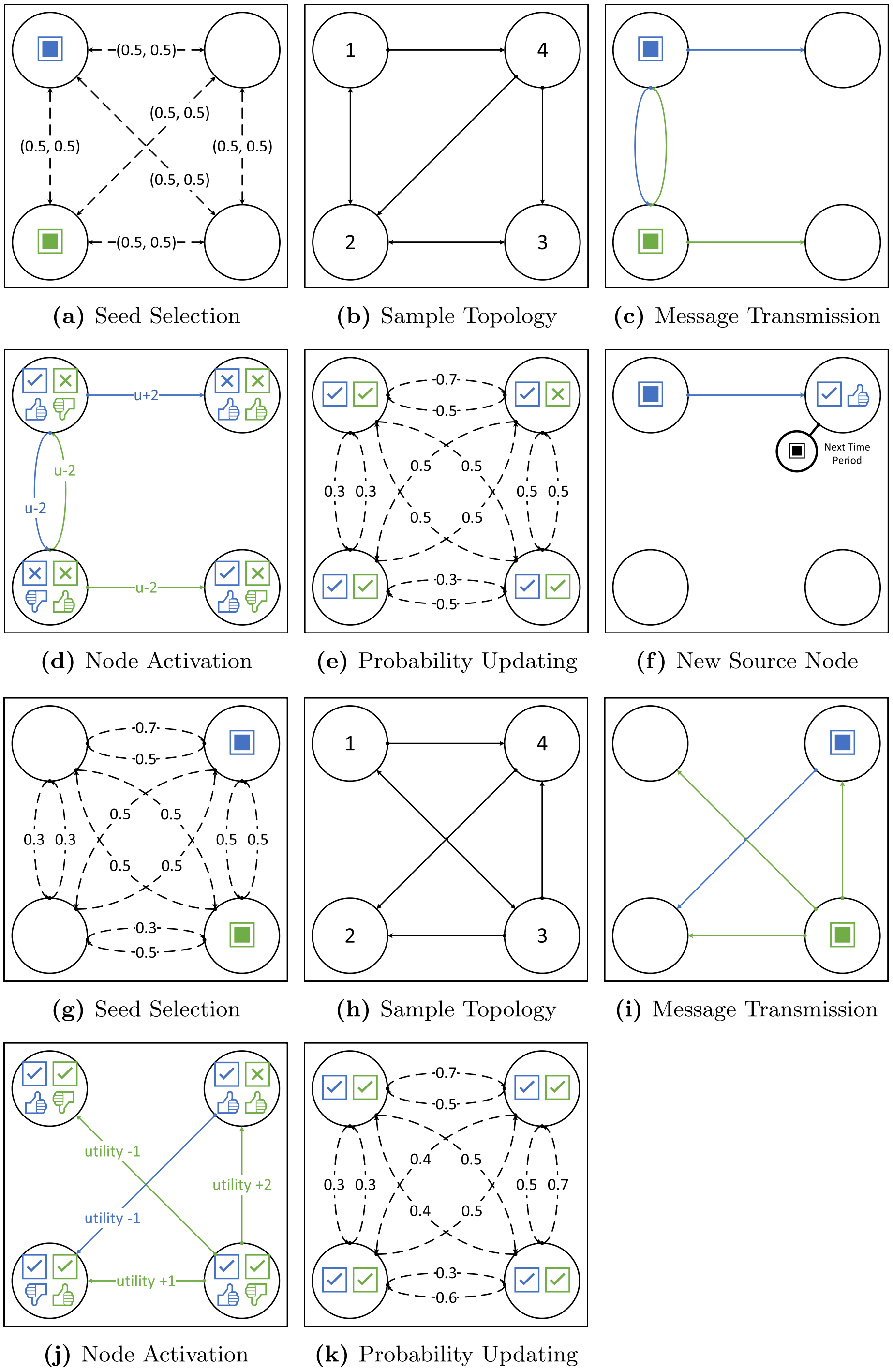
Mathematical models

In a social network, information spreads based on user-to-user interactions. Initially, some nodes will carry the designated information after being selected as seed nodes. During an independent cascade, each node plays two roles—namely, that of the message receiver, who is activated by a certain message from neighbors, and that of the message sender, who reposts the received message to their own neighbors. Information providers have several messages on hand, and they want to maximize their influence in a network. While the network users may have different preferences vis-à-vis the various messages, information providers face the problem of making the best selection of seed nodes (i.e., that which maximizes their influence).

In each period, the information provider will select the seed nodes by which to disseminate a certain message in the social network. Sometimes, it is the initial posting of a certain message, while sometimes it is a post repeated to increase network activity. Once the source user posts the message, the followers of the source users automatically receive the information. A follower make decisions based on their preferences, with different types of decisions being made as users play multiple roles in the social network (i.e., simultaneously being a follower and a followee). Information always flows from the followee to the follower, and the track of information transmission has a major influence on the network topology, where user relationships or arc connections dynamically change due to user preferences and actions. Since the Information maximization problem is subject to various uncertainties (e.g., network topology and user actions), we model this problem with stochastic programming, with the objective of maximizing the expected total influence within a finite-time horizon.

Problem description

To clearly demonstrate the information cascade process of our problem, we provide a simple example. Consider viral marketing in a random network $G(n, p)$, where a company wants to promote two products in a network that features an uncertain topology. To maximize its influence, the company wants to select certain nodes as influencers who will post the promotion message in the network. Figures 2 and 3 illustrate an example of the entire information cascade process in a four-node random network with transition probability $p = 0.5$. The symbols used in these figures are shown in Table 1. The network properties include the network sizes, transition



(a) Seed Selection (b) Sample Topology (c) Message Transmission

(d) Node Activation (e) Probability Updating (f) New Source Node

(g) Seed Selection (h) Sample Topology (i) Message Transmission

(j) Node Activation (k) Probability Updating

Fig. 3 Information cascade (time $t = 1$); information cascade (time $t = 2$)

probability, node preference, and initial activation status (Fig. 2). Figure 3 shows the dynamic network status for each information transition. The following symbols are using to explain the information cascade process.

Table 1 Notation of information cascade process

Symbol	Definition
👍	Node preference—like
👎	Node preference—dislike
☑	Node status—active
☒	Node status—inactive
▣	Seed/source node

Assume there are two message topics (i.e., BLUE and GREEN) and that the information will cascade in the random network shown in Fig. 2a, which has four nodes and whose network topology dynamically changes with initial arc probability $p = 0.5$. Before seed selection, we know about node preference in terms of message topic (Fig. 2b). Some nodes already knew of the messages before the information cascade, so we say these nodes have been "pre-activated". In Fig. 2c, nodes (1) and (4) were pre-activated by message BLUE at the initial state.

Within a single period, the information cascade usually includes four steps, as follows: seed selection, message transmission (the node sends messages), node activation (the node receives messages), and network topology probability updating. When the message provider selects the seed, the message is broadcast by the seed node in the network, but it cannot guarantee that all the other network nodes will receive the message: only followers are able to receive the message from the message sender. Following information transmission, the network topology may change. There is a strong likelihood that the link from the followee will be broken if there is a mismatch between the received message and the follower's preference. This means some directed arcs will break down, even if there were connections in the most recent time period; this is due to the uncertainty of the topology. This uncertain topology is modeled by a discrete choice model with two alternatives.

At time $t = 0$, node (1) is selected as the seed node of message BLUE, and node (2) is selected as the seed node of message GREEN. These two nodes then broadcast messages in the network. The initial probability of the directed arc connection between any two nodes is 0.5. When message transmission occurs, the real topology will be one scenario among all possibilities (Fig. 3b). The arc from node (1) to node (3) is disconnected, as is that from node (2) to node (4); this means node (3) cannot receive message BLUE and node (4) cannot receive message GREEN. Since nodes (1) and (2) are seed nodes, they alone are activated. Node (2) is activated from message BLUE by node (1). Node (2) dislikes message BLUE, and this will break the friendship between nodes (1) and (2). We leverage the utility of measuring the friendship. When the node initially receives the message, we assume it has a double effect on change to the utility. We reduce the utility from node 1 to node (2), because this is the first time node (1) receives this message. Node (4) is also activated with message BLUE from node (1). Since node (4) likes this message and had not received this message in any previous time period, node (4) will become the new source node for message BLUE and will repost message BLUE in the network (Fig. 3f). Similar to the arc utility reduction between nodes (1) and (2), the utility from node (1) to node (4) will be increased by (2) due to effective message transition.

The topology probability of the directed arc connection at the next time period is updated by the utility changing. For example, the probability of a directed arc from note 1 to node 2 is updated as

$$\begin{aligned}\text{Prob}(a_{12}^{t=1}=1) =&\text{Prob}(a_{12}^{t=1}=1|a_{12}^{t=0}=1) * \text{Prob}(a_{12}^{t=0}=1)+\\ &\text{Prob}(a_{12}^{t=1}=1|a_{12}^{t=0}=0) * \text{Prob}(a_{12}^{t=0}=0)\\ &=\frac{1}{1+exp(-(0-2))} * 0.5 + 0.5 * 0.5 = 0.31,\end{aligned}$$

where a_{12}^{t} is the directed arc connection status at time t and u_{12}^{t} is the utility at time t if $a_{12}^{t}=1$. The details of probability updating are explained in “Policy improvement” subsection.

Mathematical formulation

We formulate the Independent Cascade within Random Graph (ICRG) problem by using stochastic programming model. The authors of [14] introduce the modeling and solution to find optimal decisions in problems which involve uncertain data. In our problem, the independent cascade process include 3 decision variables, seed selection x, node activation y, message transmission z and the uncertainty is the network topology. The notation is shown in Table 2.

The original stochastic programming model [SP] is shown below:

$$[\text{SP}]\ \ \max_{x,y,z}\ \&\mathbb{E}\left(Q(x),R(y);\varepsilon\right)=\sum_{s\in\mathcal{S}}P^{s}(a)\cdot(R^{s}(y)-Q^{s}(x)), \tag{1a}$$

$$s.t.\ \&P^{s}(a)=\prod_{t\in\mathcal{T}}\prod_{i\in\mathcal{I}}\prod_{j\in\mathcal{I}\setminus\{i\}}\text{Prob}(a_{ij}^{t,s}=1)\quad\forall s\in\mathcal{S}, \tag{1b}$$

$$R^{s}(y)=\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}}w_{k}\cdot(2b_{ki}-1)\cdot(y_{ki}^{t=|\mathcal{T}|,s}-c_{ki})\quad\forall s\in\mathcal{S}, \tag{1c}$$

$$Q^{s}(x)=\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}}x_{ki}^{t,s}\quad\forall s\in\mathcal{S} \tag{1d}$$

$$x^{t,s}=x^{t,s+1}\quad\forall t\in\mathcal{T},s\in\mathcal{S}\setminus\bar{\mathcal{S}}^{t}, \tag{1e}$$

$$y_{ki}^{t,s}=\max\{c_{ki},x_{ki}^{t,s}\}\quad\forall t=0,\ k\in\mathcal{K},\ i\in\mathcal{I}, \tag{1f}$$

$$z_{ki}^{t,s}=x_{ki}^{t,s}\quad\forall t=0,\ k\in\mathcal{K},\ i\in\mathcal{I}, \tag{1g}$$

$$y_{ki}^{t,s}=\max\left\{x_{ki}^{t,s},y_{ki}^{t-1,s},\max_{j\in\mathcal{I}\setminus\{i\}}\{a_{ji}^{t,s}\cdot z_{kj}^{t-1,s}\}\right\}, \tag{1h}$$

Table 2 Notation of multistage stochastic programming model

Symbol	Definition
Indices and sets	
$i \in \mathcal{I}$	Node
$k \in \mathcal{K}$	Message
$t \in \mathcal{T}$	Time
$s \in \mathcal{S}$	Scenario
Parameters	
$a_{ij}^{t,s}$	The directed arc from node *i* to node *j*
b_{ki}	The information preference of node *i* with respect to message *k*
c_{ki}	The pre-activation, that node *i* has known or has not known the message *k* before the seed selection
w_k	The influence weight of message *k*
Decision variable	
x_{ki}^{t}	Binary variable, seed selection, whether the node *i* is selected as the seed node of message *k* at time *t*
$y_{ki}^{t,s}$	Binary variable, node activation, whether the node *i* is activated by message *k* at time *t* and scenario *s*
$z_{ki}^{t,s}$	Binary variable, message transmission, whether the node *i* decide to transmit message *k* to its neighbor at time *t* and scenario *s*
Other	
$Q(x)$	The total cost of seed selection *x*
$R(y)$	The total reward of node activation *y*
$P(a)$	The network topology probability of all the arcs *a*

$$\begin{aligned}&\forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I}\\ &z_{ki}^{t,s} = \max\left\{x_{ki}^{t,s}, b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1.s})\right\},\end{aligned} \tag{1i}$$

$$\begin{aligned}&\forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I}\\ &x \in \mathbb{B},\;\; y \in \mathbb{B},\;\; z \in \mathbb{B}.\end{aligned} \tag{1j}$$

In objective function (1a), the total influence has two parts: one is the seed cost $Q(x)$, the other one is activation reward $R(y)$. Constraint (1b) shows the probability of scenario s depend on the probability of arcs between any two nodes. The directed arc a_{ij} from node i to node j is random variable, which is following logit binary choice model with utility U_{ij}.

Utility U_{ij} is a function to measure the user friendship or the strength of arc connection, which includes two term: observed utility u_{ij} and unobserved utility ε_{ij}. The observed utility $u_{ij}^{t,s}$ at time t and scenario s is cumulative impact from node i to node j with all kinds of message topic. The current direct arc $a_{ij}^{t,s}$ from node i to node j decide the impact happen or not, the impact sign is decided by the preference b_{kj} of message k and node j, and the impact amount is decided by the transmission decision $z_{ki}^{t-1,s}$ of message k and node i at last moment. The unobserved utility $\varepsilon_{ij}^{t,s}$ is assumed to have a logistic distribution.

$$
\begin{aligned}
U_{ij}^{t,s} &= u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{U}_{ij}^{t,s} &= u_{ij}^{t-1,s} + \bar{\varepsilon}_{ij}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
a_{ij}^{t+1,s} &= \begin{cases} 1, \; U_{ij}^{t,s} > \bar{U}_{ij}^{t,s} \\ 0, \; U_{ij}^{t,s} \leq \bar{U}_{ij}^{t,s} \end{cases} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\varepsilon_{ij}^{t,s} &\sim \text{Logistic} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}
\end{aligned}
$$

Before the information cascade, there is no message transmission and each node does not know anything from the other nodes. Whether connect or disconnect, the observed utility is always be 0.

$$
\begin{aligned}
u_{ij}^{t,s} &= \bar{u}_{ij}^{t,s} = 0 \quad \forall t = -1, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
U_{ij}^{t,s} &= 0 + \varepsilon_{ij}^{t,s} \quad \forall t = -1, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{U}_{ij}^{t,s} &= 0 + \bar{\varepsilon}_{ij}^{t,s} \quad \forall t = -1, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\text{Prob}(a_{ij}^{t+1,s} = 1) &= \text{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = 0.5 \\
&\quad \forall t = -1, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}
\end{aligned}
$$

At the initial time period $t = 0$, seed node broadcast the message in the network, and some node may received message from the seed node.

$$
\begin{aligned}
u_{ij}^{t,s} &= \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot x_{ki}^{t,s} \quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{u}_{ij}^{t,s} &= 0 \quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
U_{ij}^{t,s} &= u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{U}_{ij}^{t,s} &= 0 + \bar{\varepsilon}_{ij}^{t,s} \quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\text{Prob}(a_{ij}^{t+1,s} = 1) &= \text{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = \frac{1}{1 + exp(-u_{ij}^{t,s})} \\
&\quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}
\end{aligned}
$$

From time $t = 1$ to the end of time horizon $t = \mathcal{T}$, except the seed node, the other node who received message also involve in the message transmission.

$$
\begin{aligned}
u_{ij}^{t,s} &= \sum_{\tau=0}^{t} \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{u}_{ij}^{t,s} &= \sum_{\tau=0}^{t-1} \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\Delta u_{ij}^{t,s} &= u_{ij}^{t,s} - \bar{u}_{ij}^{t,s} = \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \\
&\quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
U_{ij}^{t,s} &= u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\bar{U}_{ij}^{t,s} &= \bar{u}_{ij}^{t,s} + \bar{\varepsilon}_{ij}^{t,s} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\} \\
\text{Prob}(a_{ij}^{t+1,s} = 1) &= \text{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = \frac{1}{1 + exp(-\Delta u_{ij}^{t,s})} \\
&\quad \forall t = 0, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}
\end{aligned}
$$

The total seed cost equals to the number of seed node. The reward equals to the weighted average of final active node amount. Constraint (1c) shows the activation reward depends on message weight, node preference and node activation status y at end of the time horizon $t = |\mathcal{T}|$. Constraint (1e) is nonanticipativity constraint, and in [15] the author design an algorithm using Lagrangian dual method to solve the stochastic programming model with nonanticipativity constraint. For Constraint (1e), the scenario subset $\bar{\mathcal{S}}^t$ define as below:

$$\bar{\mathcal{S}}^t = \left\{ s \in \mathcal{S} \,|\, s = |\mathcal{S}| \cdot \frac{\tau}{|\mathcal{A}|^t} \ \forall \tau = 1, \cdots, |\mathcal{A}|^t \right\} \qquad \forall t \in \mathcal{T} \cup \{0\},$$

where the directed arc size is $\mathcal{I} \cdot (\mathcal{I} - 1)$, the combination of all arcs status is $|\mathcal{A}| = 2^{\mathcal{I} \cdot (\mathcal{I}-1)}$, and the scenario set cardinality $|\mathcal{S}| = |\mathcal{A}|^{|\mathcal{T}|} = 2^{|\mathcal{I}| \cdot (|\mathcal{I}|-1) \cdot |\mathcal{T}|}$.

The information cascade process is limited by 4 constraints. Constraints (1f, 1g) define the initial node activation and transmission decision at time $t = 0$. Constraints (1h, 1i) define the information diffusion rule from time $t = 1$ to the end $t = |\mathcal{T}|$.

In constraint (1f), some node are active node at beginning because it has already known this message c_{ki} or it is selected as seed x_{ki}. So the initial time period $t = 0$, node is not active node if and only if it did not know the message before k and it is not selected as seed node. Due to the binary property, constraint (1f) can be linearized by the equation below:

$$1 - y^t_{ki} = (1 - c_{ki}) \cdot (1 - x^t_{ki}) \quad \forall t = 0, k \in \mathcal{K}, i \in \mathcal{I}. \tag{1f-L}$$

The initial message transmission happen if and only if the node is selected as seed node, shown in constraint (1g).

Except the seed selection, the node may also be activated by two causes from time $t = 1$ to the end $t = |\mathcal{T}|$, shown in constraint (1h). One is once node i was activated by message k at previous time period $t - 1$, it will be active node in the future. The other one is at least one of the followees transmit the message k at the previous time period $t - 1$. Constraint (1h) can be linearized by the following inequalities:

$$y^{t,s}_{ki} \geq x^t_{ki} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}, \tag{1h-L1}$$

$$y^{t,s}_{ki} \geq y^{t-1,s}_{ki} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}, \tag{1h-L2}$$

$$y^{t,s}_{ki} \geq a^{t,s}_{ji} \cdot z^{t-1,s}_{kj} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}, \tag{1h-L3}$$

$$\begin{aligned} & y^{t,s}_{ki} \leq x^t_{ki} + y^{t-1,s}_{ki} + \sum_{j \in \mathcal{I} \setminus \{i\}} a^{t,s}_{ji} \cdot z^{t-1,s}_{kj} \\ & \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}. \end{aligned} \tag{1h-L4}$$

Constraint (1h-L3) is based on independent cascade assumption, that means the node will be activated ($y^{t,s}_{ki} = 1$) if the neighbor node ($a^{t,s}_{ji} = 1$) decide to transmit message ($z^{t-1,s}_{kj} = 1$). For node i, we define the number of all the neighbors as degree

$DEG_i = \sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}$. Since one of the neighbor transmit message, the receiver node will be activated, constraint (1h-L3) for all neighbor node j can be aggregated by the receiver node i.

$$y_{ki}^{t,s} \geq \frac{\sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^{t,s} \cdot z_{kj}^{t-1,s}}{\sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^{t,s}} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I} \tag{1h-L3-A}$$

Constraint (1h-L4) shows the node is deactivated if all the possible activation causes are failed.

Constraint (1i) shows node i has two motivation to transmit message k. One is node i is selected as seed, the other one is node i is new active node of message k and like this message. Constraint (1i) can be linearized by the following inequalities:

$$z_{ki}^{t,s} \geq x_{ki}^{t} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}, \tag{1i-L1}$$

$$z_{ki}^{t,s} \geq b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1,s}) \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}, \tag{1i-L2}$$

$$z_{ki}^{t,s} \leq x_{ki}^{t} + b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1,s}) \quad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I}. \tag{1i-L3}$$

Constraint (1i-L2) is based on independent cascade assumption, that means the node is willing to transmit message ($z_{ki}^{t,s} = 1$) if it likes this message ($b_{ki} = 1$) and it just activated ($y_{kj}^{t,s} = 1$) and never knew this message before ($y_{kj}^{t-1,s} = 0$). Constraint (1i-L3) shows the node decided not to transmit the message if all the transmission motivations are invalid.

The computation complexity of this model is $O(2^{|\mathcal{K}||\mathcal{I}| \cdot log_{|T|}|S| \cdot |T|})$. To reduce the complexity, we add an assumption of seed selection, that the decision-maker only allows to select one seed node of each message within one time period. It is formulated by the following constraint:

$$\sum_{i \in \mathcal{I}} x_{ki}^{t,s} = 1 \;\; \forall s \in \mathcal{S}, t \in \mathcal{T} \cup \{0\}, k \in \mathcal{K}. \tag{1d-A}$$

The computation complexity is reduced to $O(|\mathcal{I}|^{|\mathcal{K}| \cdot log_{|T|}|S| \cdot |T|})$ after adding this assumption, and the objective function (1a) can be simplified as below:

$$\begin{aligned} \max_{x,y,z} \; \mathbb{E}\,(Q(x), R(y); \varepsilon) &= \sum_{s \in \mathcal{S}} P^s(a) \cdot (R^s(y) - Q^s(x)) \\ &= -|\mathcal{T} + 1| \cdot |\mathcal{K}| + \sum_{s \in \mathcal{S}} P^s(a) \cdot R^s(y)). \end{aligned} \tag{1a-A}$$

Solution approaches

Since the network topology is dynamically changed, the decision-maker is faced with an unstable node friendship. The uncertain directed arc connection leads to the scenario size exponentially growth with the network size $|I|$ and time horizon $|T|$. To handle the large-scale scenarios, we have two approaches to solve the information cascade in random graph problem:

- Myopic policy: does not explicitly use any forecasted network topology and separate the multistage into several two-stage problems (MYSP) by discrete time.
- Reinforcement learning: reformulates the stochastic programming model to Markov decision process (MDP)

Two-stage stochastic programming with myopic policy

Contrary to the original model, the myopic model focuses on current network topology and ignores the future changing on arc. The seed selection (x^t) is only based on current user connection (a^t) and aims to find the local maximal influence on node activation of next time period (y^{t+1}):

$$x^t = \arg\max R(y^{t+1}, a^t).$$

By using the myopic method, the multistage problem is decomposed to several two-stage problems. The first stage variable is seed selection, and the second stage variable is node activation and node repost decision. The given parameters include the node preference, the probability of current network, and the node repost decision of the previous time period. Since we select seed to find the maximal expected influence at current time period, the decision only happens within one time period. Then the time index and set can be removed and the node repost decision of the previous time period should be added in the known parameter. The notation of myopic model is shown in Table 3.

The mathematical formulation of myopic model is shown below:

$$[\text{MYSP}] \quad \max_{x,y} \ \&\mathbb{E}(R(y); \varepsilon) = \sum_{s \in \mathcal{S}} P^s(a) \cdot R^s(y), \tag{2a}$$

$$s.t. \ \&P^s(a) = \prod_{i \in \mathcal{I}} \prod_{j \in \mathcal{I} \setminus \{i\}} \text{Prob}(a^s_{ij} = 1) \quad \forall s \in \mathcal{S}, \tag{2b}$$

$$R^s(y) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} w_k \cdot (2b_{ki} - 1) \cdot (y^s_{ki} - c_{ki}) \quad \forall s \in \mathcal{S}, \tag{2c}$$

$$\sum_{i \in \mathcal{I}} x_{ki} = 1 \quad \forall k \in \mathcal{K}, \tag{2d}$$

$$y^s_{ki} \geq c_{ki} \quad \forall s \in \mathcal{S}, \, k \in \mathcal{K}, \, i \in, \mathcal{I}, \tag{2e}$$

$$y^s_{ki} \geq x_{ki} \quad \forall s \in \mathcal{S}, \, k \in \mathcal{K}, \, i \in \mathcal{I}, \tag{2f}$$

Table 3 Notation of myopic two-stage stochastic programming model

Symbol	Definition
Indices and sets	
$i \in \mathcal{I}$	Node
$k \in \mathcal{K}$	Message
$s \in \mathcal{S}$	Scenario
Parameters	
a_{ij}^{s}	The directed arc from node *i* to node *j*
b_{ki}	The information preference of node *i* with respect to message *k*
c_{ki}	The pre-activation, that node *i* has known or has not known the message *k* before the seed selection
d_{ki}	The node repost decision, that node *i* will repost message *k* in the network
w_k	The influence weight of message *k*
Decision variable	
x_{ki}	Binary variable, seed selection, whether the node *i* is selected as the seed node of message *k* at time *t*
y_{ki}^{s}	Binary variable, node activation, whether the node *i* is activated by message *k* at time *t* and scenario *s*

$$y_{ki}^{s} \geq \frac{\sum\limits_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^{s} \cdot (d_{ki} + x_{kj} - d_{ki} \cdot x_{kj})}{\sum\limits_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^{s}}, \quad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I}, \tag{2g}$$

$$y_{ki}^{s} \leq c_{ki} + x_{ki} + \sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^{s} \cdot (d_{ki} + x_{kj} - d_{ki} \cdot x_{kj}), \quad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I}, \quad x \in \mathbb{B},\ y \in \mathbb{B},\ z \in \mathbb{B}. \tag{2h}$$

When time $t > 0$, some known parameters is given by the previous myopic model.

$$\begin{aligned}
c_{ki} &= \hat{y}_{ki} \quad \forall\, k \in \mathcal{K},\, i \in \mathcal{I},\\
d_{ki} &= b_{ki} \cdot (\hat{y}_{ki} - \hat{c}_{ki}) \quad \forall\, k \in \mathcal{K},\, i \in \mathcal{I},\\
u_{ij} &= \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot \hat{a}_{ij} \cdot (\hat{d}_{ki} + \hat{x}_{kj} - \hat{d}_{ki} \cdot \hat{x}_{kj}) \quad \forall\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\},\\
\mathrm{Prob}(a_{ij}^{s} = 1) &= \frac{1}{1 + exp(-u_{ij})} \quad \forall\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\},
\end{aligned}$$

where $\hat{y}_{ki}$ is the activation status using the decision of previous seed selection $\hat{x}_{kj}$, $\hat{c}_{ki}$ is the parameter of previous myopic model, and $\hat{d}_{ki}$ is the node repost decision using the decision of previous seed selection $\hat{x}_{kj}$. The parameter transition between two myopic models is shown in Fig. 4.

Reinforcement learning with Markov decision process

Our problem can be defined as a Markov decision process (MDP), that how information provider chooses a source user when facing the given information activation status of all users in the network. We use the Reinforcement Learning to learn the policy based on state-action pairs $(\boldsymbol{s}, \boldsymbol{a})$. The notation of reinforcement learning with Markov decision process model is shown in Table 4.

In general, MDP is described by a 4-tuple (S, A, P, R), which are the states, actions, transitions, and reward. In our problem, these four terms are defined as below:

- S: the finite set of state, i.e., activation status, $\boldsymbol{s} \in S$
- A: the finite set of action, i.e., source user selection, $\boldsymbol{a} \in A$
- P: the probability of transition from s to s' through action a, $P_a(\boldsymbol{s}, \boldsymbol{s'})$
- R: the expected reward of transition from s to s' through action a, i.e., weighted information influence, $R_a(\boldsymbol{s}, \boldsymbol{s'})$.

The probability function is not unknown since the network topology is uncertainty. The reward function is shown below:

$$R(\boldsymbol{s}, \boldsymbol{s'}) = \sum_{k \in K} \sum_{i \in I} w_K \cdot (s'_{ki} - s_{ki}). \tag{3a}$$

We will introduce the Q-learning algorithm to compute optimal policies, which includes policy evaluation and policy improvement.

Policy evaluation

If we have a policy, the probability of actions taken at each state are known. Then the MDP is turned into a Markov chain (with rewards). We can compute the expected total reward collected over time using this policy. For given policy $\pi(\boldsymbol{s})$, the state-value function $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ is used the evaluated the policy value.

$$Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}^{\pi}\left(R(\boldsymbol{s}, \boldsymbol{s'}) + \gamma \cdot \sum_{\boldsymbol{a'} \in A} \pi(\boldsymbol{s'}, \boldsymbol{a'}) \cdot Q^{\pi}(\boldsymbol{s'}, \boldsymbol{a'})\right) \forall\, \boldsymbol{s} \in S,\ \boldsymbol{a} \in A, \tag{3b}$$

where γ is the discount factor and $\pi(\boldsymbol{s}, \boldsymbol{a})$is the probability to take action $\boldsymbol{a}$ at state $\boldsymbol{s}$.

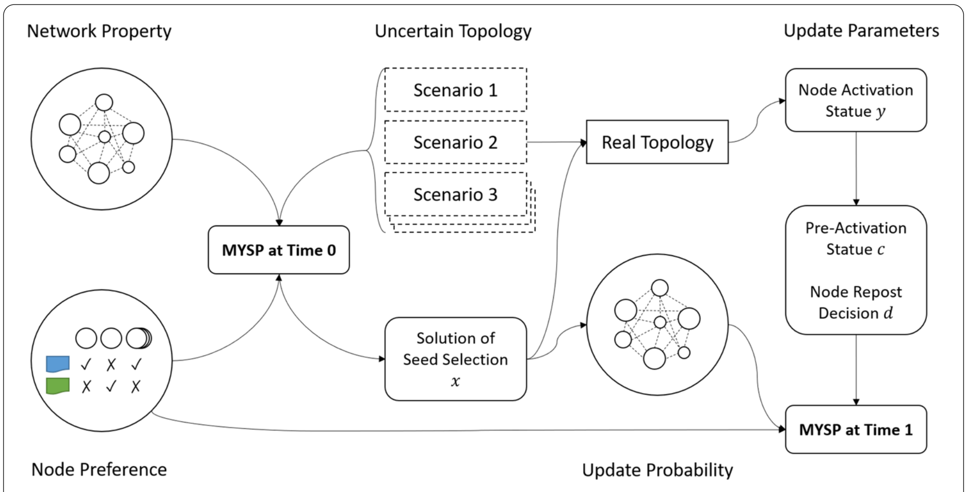


Fig. 4 Myopic model: parameter transition

Table 4 Notation of reinforcement learning with Markov decision process model

Symbol	Definition
Indices and sets	
$i \in \mathcal{I}$	Node
$k \in \mathcal{K}$	Message
Parameters	
b_{ki}	The information preference of node *i* with respect to message *k*
w_k	The influence weight of message *k*
ρ_{ij}	The probability of arc connection from node *i* to node *j*
Variable	
σ_{ki}	The element of state matrix $\boldsymbol{s} \in S$ in row *k* and column *i*, that the activation status of node *i* by message *k*
$\boldsymbol{\alpha}_{ki}$	The element of action matrix $\boldsymbol{a} \in A$ in row *k* and column *i*, that the seed selection of node *i* by message *k*

Consider a network with node size $|\mathcal{I}| = 4$ and information size $|\mathcal{K}| = 2$. The size of state set is $|S| = 2^{|\mathcal{K}|\cdot|\mathcal{I}|} = 256$ and the size of action set $|A| = |\mathcal{I}|^{|\mathcal{K}|} = 16$. Given initial state (no activation) $\boldsymbol{s}$, the information provider has a trivial policy $\pi(\boldsymbol{s})$, that each node has equally probability to be seed.

$$\boldsymbol{s} = \left.\left(\underbrace{\begin{matrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \end{matrix}}_{\text{user } i}\right)\right\}\text{information } k = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\pi(\boldsymbol{s}) = \left.\left(\underbrace{\begin{matrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \end{matrix}}_{\text{user } i}\right)\right\}\text{information } k = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

We run several simulations of independent cascade with random actions and discount factor $\gamma = 1$. The simulation uses the Monte Carlo method. Before cascade starting, we generate a large number of pseudo-random uniform variables from the interval [0,1], which is used to decide the network topology. If the value falls into the probability interval of the $\pi(\boldsymbol{s}, \boldsymbol{a})$, we will take action a when we meet state s. For example, we have 3 options for state s, action a_1, a_2, a_3, and the probability to take the actions are $\pi(s, a_1) = 0.1, \pi(s, a_2) = 0.3, \pi(s, a_3) = 0.6$. Based on the definition of Monte Carlo method, there are 3 probability intervals [0, 0.1], (0.1, 0.4], (0.4, 1] respect to different action. When we got the random number 0.5, it falls into the probability interval (0.4, 1], which means we will take action a_3.

The average final influence of each action is shown in Table 5. Figure 5 shows the same policy is applied in different state to calculate the expected total reward, that is the total activated node at end of the time horizon.

Policy improvement

Based on the simulation result, we create a final reward (weighted total influence) list $Q(\boldsymbol{s}, \boldsymbol{a})$ by state and action, which is used to improve the policy. $\pi(\boldsymbol{s}, \boldsymbol{a})$ and $\pi'(\boldsymbol{s}, \boldsymbol{a})$ are old policy and new policy. The action set A is splitted to two subset. A^1 is the set of all happened action, A^0 is the set of all unhappened action:

$$\pi'(\boldsymbol{s},\boldsymbol{a}) = \begin{cases} (1 - \sum\limits_{\boldsymbol{a}\in A^0} \pi(\boldsymbol{s},\boldsymbol{a})) \cdot \dfrac{Q(\boldsymbol{s},\boldsymbol{a}) - \hat{Q}(s,a)}{\sum\limits_{a\in A^1} Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) - \hat{Q}^{\pi}(\boldsymbol{s},\boldsymbol{a})}, & \forall\, \boldsymbol{a} \in A^1,\, \boldsymbol{s} \in S \\ \pi(s,a), \quad \forall\, \boldsymbol{a} \in A^0,\, \boldsymbol{s} \in S \end{cases}$$

$$\hat{Q}(\boldsymbol{s},\boldsymbol{a}) = \lambda \cdot \min_{\boldsymbol{a}\in A^1} Q(\boldsymbol{s},\boldsymbol{a}),$$

where λ is the stepsize, which is decided by the iteration number and policy improved value.

$$\lambda = \frac{m}{itr^n} \cdot \sum_{\boldsymbol{s}\in S} \sum_{\boldsymbol{a}\in A} \left(\pi_{itr}(\boldsymbol{s},\boldsymbol{a}) \cdot Q^{\pi_{itr}}(\boldsymbol{s},\boldsymbol{a}) - \pi_{itr-1}(\boldsymbol{s},\boldsymbol{a}) \cdot Q^{\pi_{itr-1}}(\boldsymbol{s},\boldsymbol{a}) \right)$$

For the example of Policy Evaluation, the updated policy is shown in Table 6. If we summarized the policy by information k and user i, it will be

$$\pi(\boldsymbol{s}) = \left.\underbrace{\begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \end{pmatrix}}_{\text{user } i}\right\}\text{information } k = \begin{pmatrix} 0.2056 & 0.2086 & 0.2959 & 0.2899 \\ 0.2255 & 0.2505 & 0.2697 & 0.2542 \end{pmatrix}$$

Computational experiments on algorithms' convergence

Numerical experiments and results of different algorithms are presented in this section on solving the information maximization problem. We randomly generate and test three data sets, i.e.,

- data set (2,4) with 2 messages and 4 nodes
- data set (2,7) with 2 messages and 7 nodes
- data set (3,7) with 3 messages and 7 nodes (cannot converge within 24 h, the average computation time for each iteration will take 60000 s).

Table 5 Example of policy evaluation

State	Action	Influence
s	a	$Q^{\pi}(s,a)$
$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 1&0&0&0 \\ 1&0&0&0 \end{pmatrix}$	3.27869
	$\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \end{pmatrix}$	3.09836
	$\begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \end{pmatrix}$	3.22414
	$\vdots$	$\vdots$
	$\begin{pmatrix} 0&0&0&1 \\ 0&0&0&1 \end{pmatrix}$	3.90909

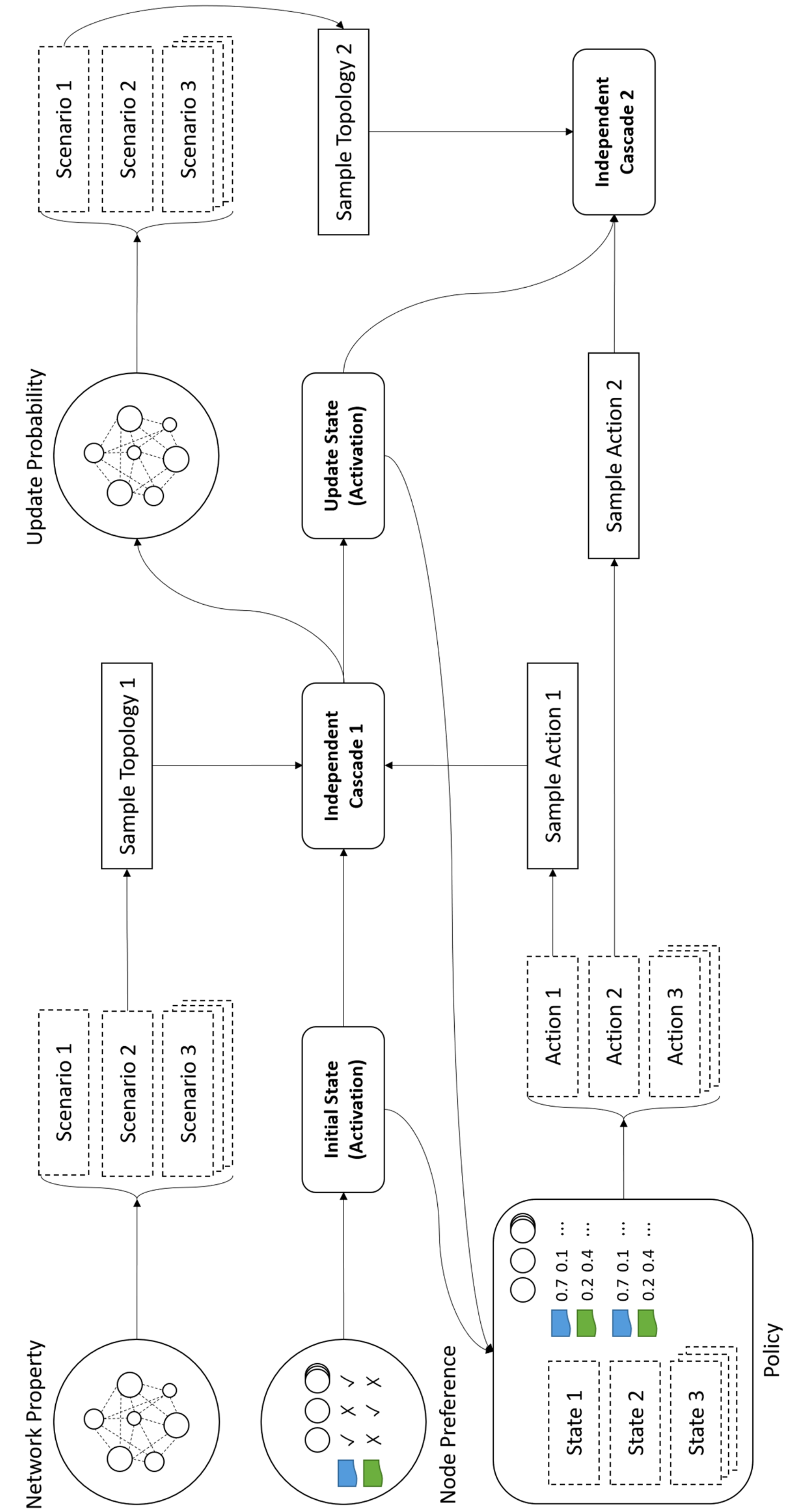


Fig. 5 Reinforcement learning: policy evaluation

Table 6 Example of policy improvement

State	Action	Initial policy	Updated policy
s	a	$\pi(s,a)$	$\pi'(s,a)$
$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$	0.0625	0.0463788
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	0.0625	0.0515202
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	0.0625	0.0554653
	$\vdots$	$\vdots$	$\vdots$
	$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	0.0625	0.0737093

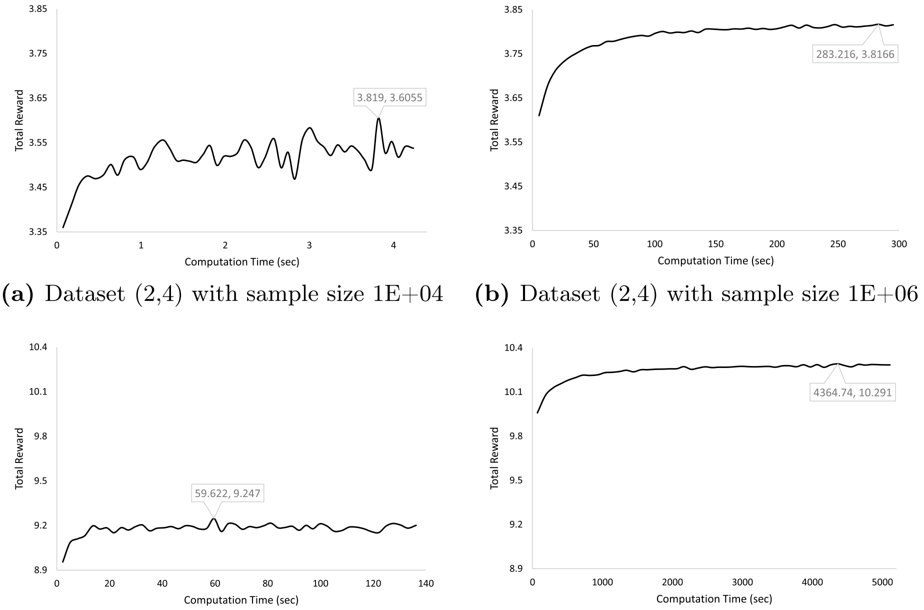
Table 7 Computation results

Method	Best result	Time (s)	Best result	Time (s)
	Dataset (2,4)		Dataset (2,7)	
SP-MYOPIC	3.771	5.349	10.238	69.701
RL-MDP sample size 1E+04	3.6055	3.819	9.247	59.622
RL-MDP sample size 1E+06	3.8166	283.216	10.291	4364.74

The algorithms are coded in Microsoft Visual Studio 2019 C++ linked with CPLEX 12.9. All the programs are run on the Microsoft Windows 10 Professional operating system with Intel Xeon CPU E-2186 2.90GHz and 32GB RAM. Since the computation of data set (3,7) cannot converge within practical time, we will only discuss the results of data set (2,4) and data set (2,7). All the computation results are shown in Table 7.

In Fig. 6, we show the results of total rewards when we increase the number of iterations while implementing the algorithm of reinforcement learning with Markov decision process. In Fig. 6a, b, the results are based on experiments with 2 messages and 4 nodes, while a 7-node case is shown in Fig. 6c, d. In all the figures, the left sides show the convergence results using a sample size of 10,000 for the Monte Carlo simulation in the algorithm, and with 1 million for the right sides. It can be easily observed that a larger sample size can converge faster and achieve policies with higher objective value in short amount of time. This is partially due the fact that a smaller sample size provides lower accuracy in approximation.

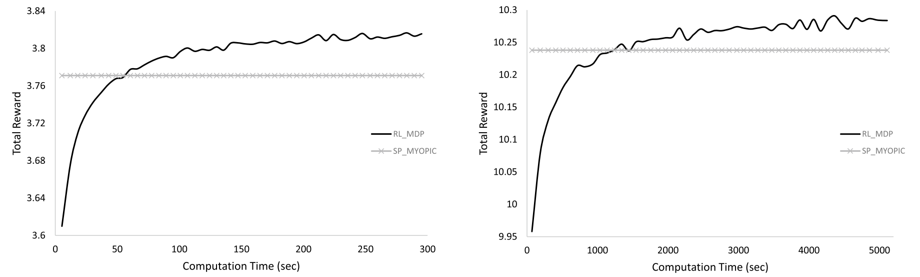
In Fig. 7, we compare the two proposed algorithms, i.e., the two-stage stochastic programming with Myopic policy (SP-MYOPIC) and the algorithm via reinforcement learning with Markov decision process (RL-MDP) using the different data set. In both cases (2 messages plus 4 nodes versus 2 messages plus 7 nodes), we use a sample size of 1 million. The SP-MYOPIC approach's result is the straight, horizontal line in both sub-figures. Although it is faster to calculate and does not have convergence issues, it



(a) Dataset (2,4) with sample size 1E+04 **(b)** Dataset (2,4) with sample size 1E+06

(c) Dataset (2,7) with sample size 1E+04 **(d)** Dataset (2,7) with sample size 1E+06

Fig. 6 Computation result using RL-MDP



(a) Dataset (2,4) with sample size 1E+06 **(b)** Dataset (2,7) with sample size 1E+06

Fig. 7 Algorithm comparison

is trailing after the RL-MDP method in terms of total rewards when a certain amount of computational time is provided.

Conclusion

In this study, we presented multistage stochastic mixed integer nonlinear programming models with endogenous uncertainty to examine influence maximization in social networks that feature a dynamic topology decided by users. We proposed two methods, each featuring a network structure based on user preference in a finite-time information cascade. One makes use of classic two-stage stochastic programming, while the other leverages reinforcement learning. Information networks generally comprise autonomous nodes that make decisions when forming links with other nodes and transmitting information. We used the discrete choice model to build the node preference distribution; additionally, we modeled dynamic changes to the network structure by using stochastic dynamic programming, which can be solved via the Markov decision process. Our models accurately describe and predict user behavior so as to ensure dynamic optimization under uncertainty; as such, they act as tools by which to analyze dynamic changes to network structure by controlling information flow, and can be used in the information maximization problem. The results of our computational experiments show that large sample sizes can provide better and more stable results when one implements the reinforcement-learning based approach, which performs better than the two-stage stochastic programming (i.e., myopic) approach.

Acknowledgements
This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute.

Further information
A preliminary version of this journal paper appeared as a proceeding article in: Tagarelli A., Tong H. (eds) Computational Data and Social Networks. CSoNet 2019. Lecture Notes in Computer Science, Vol. 11917. The current journal paper has since improved substantially.

Authors' contributions
MC conducted the experiments. All authors contributed to the analysis of the results and to writing the paper. All authors read and approved the final manuscript.

Funding
The work was supported in part by the U.S. Air Force Research Laboratory (AFRL) award FA8651-16-2-0009.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Department of Industrial Engineering and Management Systems, University of Central Florida, 12800 Pegasus Drive, Orlando, FL, USA. [2] Air Force Research Laboratory, Eglin AFB, FL, USA.

Received: 7 April 2020 Accepted: 27 January 2021

Published online: 22 February 2021

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. New York: ACM; 2003. p. 137–46.
2. Guille A, Hacid H, Favre C, Zighed DA. Information diffusion in online social networks: a survey. ACM Sigmod Rec. 2013;42(2):17–28.
3. Goldenberg J, Libai B, Muller E. Talk of the network: a complex systems look at the underlying process of word-of-mouth. Market Lett. 2001;12(3):211–23.
4. Granovetter M. Threshold models of collective behavior. Am J Sociol. 1978;83(6):1420–43.
5. Saito K, Nakano R, Kimura M. Prediction of information diffusion probabilities for independent cascade model. In: International conference on knowledge-based and intelligent information and engineering systems. Berlin: Springer; 2008. p. 67–75.
6. 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. New York: ACM; 2010. p. 1029–38.
7. Wang C, Chen W, Wang Y. Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining Knowl Discov. 2012;25(3):545–76.
8. Oinas-Kukkonen H. A foundation for the study of behavior change support systems. Personal Ubiquitous Comput. 2013;17(6):1223–355.
9. Ploderer B, Reitberger W, Oinas-Kukkonen H, Gemert-Pijnen J. Social interaction and reflection for behaviour change. Personal Ubiquitous Comput. 2014;18(7):1667–766.
10. Yu L, Cui P, Wang F, Song C, Yang S. From micro to macro: uncovering and predicting information cascading process with behavioral dynamics. arXiv preprint; 2015. arXiv:1505.07193.
11. Myers SA, Leskovec J. The bursty dynamics of the twitter information network. In: Proceedings of the 23rd international conference on world wide web; 2014. p. 913–24.

12. McFadden D. Conditional logit analysis of qualitative choice behavior. In: Zarembka P, editor. Frontiers in econometrics, Chap. 4. New York: Academic Press; 1973. p. 105–142.
13. McFadden D. Econometric models of probabilistic choice. Structural analysis of discrete data with econometric applications. 1981. 198272.
14. Birge JR, Louveaux F. Introduction to stochastic programming. Berlin: Springer; 2011.
15. Escudero LF, Garín MA, Pérez G, Unzueta A. Scenario cluster decomposition of the lagrangian dual in two-stage stochastic mixed 0–1 optimization. Comput Oper Res. 2013;40(1):362–77.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.