Applied Intelligence (2019) 49:2137–2150
https://doi.org/10.1007/s10489-018-1387-8

Community-based influence maximization for viral marketing

Huimin Huang[1] · Hong Shen[1,2] · Zaiqiao Meng[1] · Huajian Chang[1] · Huaiwen He[1]

Published online: 2 January 2019

Abstract

Derived from the idea of word-to-mouth advertising and with applying information diffusion theory, viral marketing attracts wide research interests because of its business value. As an effective marketing strategy, viral marketing is to select a small set of initial users based on trust among close social circles of friends or families so as to maximize the spread of influence in the social network. In this paper, we propose a new community-based influence maximization method for viral marketing that integrates community detection into influence diffusion modeling, instead of performing community detection independently, to improve the performance. We first build a comprehensive latent variable model which captures community-level topic interest, item-topic relevance and community membership distribution of each user, and we propose a collapsed Gibbs sampling algorithm to train the model. Then we infer community-to-community influence strength using topic-irrelevant influence and community topic interest, and further infer user-to-user influence strength using community-to-community influence strength and community membership distribution of each user. Finally we propose a community-based heuristic algorithm to mine influential nodes that selects the influential nodes with a divide-and-conquer strategy, considering both topic-aware and community-relevant to enhance quality and improve efficiency. Extensive experiments are conducted to evaluate effectiveness and efficiency of our proposals. The results validate our ideas and show the superiority of our method compared with state-of-the-art influence maximization algorithms.

Keywords Social networks · Viral marketing · Influence maximization · Latent variable model

1 Introduction

Due to its main application in business, i.e., word-of-mouth or viral marketing, influence maximization has attracted wide research interests in recent years. Influence maximization aims to solve the problem of selecting a fixed size set of seed nodes in a social network to maximize the influence spread. Domingos and Richardson [1, 2] proposed an influence diffusion model in social networks based on Markov Random Field in 2001, when influence diffusion model was first introduced into the field of computer science. Kempe et al. [3] proposed Linear Threshold (LT) and Independent Cascade (IC) to model the process of influence diffusion, proved influence maximization in LT and IC is a NP-hard problem, and proposed a greedy algorithm with an approximate degree of $(1 - 1/e)$ to solve the problem. Such models summarize previous research experience in statistical physics, social psychology and marketing, and maintain good properties in process of information diffusion, such as monotonicity and submodeling.

Influence maximization problem has been extensively studied and several influence maximization algorithms have been proposed including [3–12]. Thereinto, algorithms in [3, 5, 7, 8, 11] are based on greedy. They take full advantage of submodularity and monotony of IC model or LT model, and to evaluate the influence, they utilize Monte Carlo simulations, which can ensure the accuracy of the influence spread value. However, tens of thousands of Monte Carlo simulations in large-scale social networks would greatly reduce the efficiency of algorithms. Algorithms in [4, 6, 9–12] emphasize analyzing and exploiting topology structure of network to improve the algorithm, and they achieve much higher execution efficiency than greedy algorithms.

Recently, community-based influence maximization algorithms are proposed. As community-based methods

✉ Hong Shen
shenh3@mail.sysu.edu.cn

Huimin Huang
huanghm45@gmail.com

[1] School of Data and Computer Science,
Sun Yat-sen University, Guangzhou, China

[2] School of Computer Science, University of Adelaide,
Adelaide, Australia

find seed node set within the communities instead of over the whole network, these algorithms are generally more efficient than traditional greedy algorithms and suitable to be performed parallelized. However, there are still large drawbacks in the existing community-based influence maximization algorithms. Firstly, the existing algorithms usually consist of two steps: community detection and seed selection that causes non-trivial complexity and limits their efficiency and scalability. Secondly, some of them assume influence strength between users is known, which may be infeasible because, in some cases, influence strength between users are unobservable. And some of them compute influence strength between users depending only on social relationships, overlooking topic distribution on the social links which is exactly crucial to improve the accuracy of prediction. Thirdly, they tend to select uniform seed node set for promoting different items that may be unreasonable because in the real word, different items have different topic distribution.

To overcome above limitations, we propose **C**ommunity-based **T**opic-aware **I**nfluence **M**aximization **(CTIM)**, which is topic-aware and finds influential nodes with regards to communities, integrating influence strength computation with community detection in a comprehensive latent variable model, through which user topic relevance and item topic relevance can be extracted effectively. Our motivation is from the observations of marketing and psychology that (1) Friend relationship influences user making decision on item selection; (2) The process of influence propagation can reveal users' preference; (3) Different items have different characteristics. Thus, we first design a comprehensive latent variable model to learn user interest and item characteristics as well as perform community detection, jointly applying influence diffusion modeling and probabilistic topic modeling; then propose a community-based influence maximization algorithm to find the influential nodes.

We integrate the community detection into the influence diffusion modeling instead of performing community detection independently, which improves the efficiency of the algorithm largely. We assume that each user may behave as the member of different communities in different settings. Thus, we denote each user to belong to different communities with a multinomial distribution. Through our latent variable model, community membership distribution of each user can be inferred, and then community detection can be achieved. For example, assuming we have inferred user v's community membership distribution $\left\{\boldsymbol{\pi}_{vc_1}, \boldsymbol{\pi}_{vc_2}, ...\right\}$ and $\boldsymbol{\pi}_{vc_j}$ is the maximum of community membership value, user v would be divided into community c_j.

We argue that purchase behaviors are relevant to communities' interests. Fundamentally, community is a group of users that have denser connections among the group than with the rest of the network [13]. Users engage in social network as members of communities. Lots of previous researches have disclosed that users in communities share common properties or attributes [14–16]. In other words, communities have their attributes including community interest etc. Assuming each user may behave as the member of different communities in different settings, user interest is interrelated with community interest by the community membership distribution of the user. As is known that purchase behaviors are relevant to user interest. In consequence, purchase behaviors are related with community interest. Besides, studies on sociology and marketing demonstrate that the behaviors of some individual users are highly volatile [17], which makes it difficult to accurately mine user preferences and diffusion properties at individual level. While, community-level model not only can avoid volatility of individual users but also alleviate the problem of data sparsity. Thus we utilize the compact community-level extraction to capture user interest more effectively.

Community-based topic-aware influence maximization faces several challenges. Firstly, communities and topic distribution over users and items are hidden, which are three critical factors in the process of influence diffusion. We need to model the correlation between them suitably. Secondly, we would like to integrate the process of community detection into the influence diffusion modeling to improve the efficiency of the algorithm. Thirdly, influence strength between users is unknown. Besides, we are required to select different seed set to prompt different items since different items have different topic distribution.

To sum up, our major contributions in this paper are:

(1) We propose a new community-based influence maximization method, which mines latent topic information as well as community memberships in the process of influence propagation, thereby infers influence strength between users and finds influential nodes with regards to communities.
(2) We develop a comprehensive latent variable model. It infers item topic relevance and captures user interest across communities avoiding volatility of individual users and alleviating problem of data sparsity. It integrates the community detection into the influence diffusion modeling that improves the efficiency of the algorithm largely.
(3) We propose a community-based influence maximization algorithm which is also topic-aware, therefore it has superiority in both efficiency and accuracy of prediction.
(4) We provide a thorough analysis of our influence maximization method, and perform experiments to demonstrate its effectiveness and efficiency compared with the state-of-the-art algorithms.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the preliminaries including notation, terminology and our task. Section 4 details the proposed method. Section 5 reports our experimental evaluation. Section 6 concludes the paper.

2 Previous work

2.1 Influence maximization

Following the general Greedy algorithm proposed by Kempe et al. [3] to solve the problem of maximizing influence, Leskovec et al. [7] propose the CELF algorithm, which is based on the submodularity of the objective function in LT and IC models and employs lazy-forward optimization. The idea behind CELF is that the marginal gain provided by a node in the current iteration cannot be better than the marginal gain provided by the node in the previous iteration. Compared with the general Greedy algorithm, this method greatly improves the efficiency by nearly 700 times.

Goyal et al. [5] propose a CELF++ algorithm based on CELF algorithm, which improved the efficiency of the original algorithm (CELF) by 35%–55%. Chen et al. [11] propose the NewGreedy and MixedGreedy algorithms using the idea of reducing the scale of the propagation map. The NewGreedy algorithm obtains a subgraph by deleting each edge in the original image with a certain probability, and calculates the influence spread of the node in the subgraph. The MixedGreedy algorithm adds CELF improvements based on the NewGreedy algorithm. Since these two algorithms still require a lot of simulations, they cannot be applied to large-scale networks. Cheng et al. [8] propose the StaticGreedy algorithm, which guarantees the optimization of the submodularity of the target by repeatedly using the generated subgraph, and improves the efficiency of the algorithm by two orders of magnitude. Luo et al. [18] use PageRank to evaluate the influence of nodes firstly, and select a certain percentage of nodes as candidate seed nodes to improve the efficiency of the algorithm.

Depth-based heuristic algorithm [11], PMIA [4], MIA [19] are proposed, which estimate the node influence through the path function. Thereinto, MIA algorithm estimates the influence from node u to node v using maximum influence path (MIP), i.e., the path with maximum activation probability, and sets a probability threshold θ for MIP. The algorithm builds maximum influence in-arborescence of a target node v by assembling all the maximum influence paths from source nodes to target node v, obtains activated probability of the node v through recursive computations, and computes influence spread of seed set by summing the activated probabilities of all nodes in the network.

Tang et al. proposes TIM+ [9] and IMM [10], which sample nodes from the network propagation graphs to establish a reverse reachable set (Reverse Reachable). If the number of nodes in the RR set is huge, the influence of the node is also huge. The DegreeDiscount algorithm [11] evaluates the influence of nodes based on the number of degrees.

Cao et al. [20] first propose the community-based influence maximization algorithm OASNET (OptimalAllocation in a Social NETwork). The algorithm assumes that different communities are independent of each other and influence cannot be transmitted between communities. The community structure is detected by the CNM (Clauset-Newman-Moore) algorithm [21]. Wang et al. propose community-based greedy algorithm CGA with a provable approximation guarantee [22]. Chen et al. [23] use the heat diffusion model to study the community-based influence maximization problem and propose the Community-based Influence Maximization (CIM) algorithm. Li et al. [24] consider the conformity of nodes and propose the community-based influence maximization algorithm CINEMA (Conformity-aware Influence Maximization). Community-based influence maximization algorithms are generally more efficient than traditional greedy algorithms, and suitable to be performed parallelized.

2.2 Probabilistic topic modeling

Probability topic modeling is aimed at finding the topic structure hidden in the massive documents. Since Latent Dirichlet Allocation (LDA) was first proposed by Blei in 2003 [25], probability topic model has been widely studied in the field of probability semantic analysis. Yan et al. [26] proposes a short text modeling method, biterm topic model (BTM), which directly models the generation of word co-occurrence patterns (i.e. biterms) in the whole corpus. Yin and Wang [27] applies a Dirichlet multinomial mixture model-based approach for short text clustering. Yang et al. [28] incorporates demographic information of review authors into topic modeling to accomplish review sentiment classification and user attribute prediction. Many topic-based detection models by LDA have already been proposed including topic change detection model [41], semantically enhanced document retrieval model [42], LDA and Linear Algebra based LSA combination model [43], GSLDA model for group spamming detection [44]. Beside, the dynamic topic model has been proposed including online multi-scale dynamic topic model [29], dynamic clustering of streaming short documents [30], dynamic user clustering topic model [31]. With the high popularity of social medias, recently some works propose to integrates content topic discovery and social influence analysis in the same generative process. Representative models include extension of LDA model to handle popular nodes in social networks [32], social-relational topic model [33], Followship-LDA (FLDA) model [34].

2.3 Social influence diffusion

Initially motivated by viral marketing, the problem of social influence diffusion emerges which increasingly concerns the propagation of ideas, opinions or rumors etc. through social networks. Barbieri et al. [35] proposes Topic-aware Independent Cascade (TIC) Model, where the user-to-user influence probabilities depend on the topics, and an Expectation-Maximization method is used to learn parameters, but the semantic information in the networks has not been utilized, thereby missing important opportunities to better capture the properties of influence propagation. Chen et al. [36] considers a deadline constraint to reflect the time-critical effect in influence diffusion and develops a new propagation model, independent cascade model with meeting events to capture the delay of propagation in time. Hu et al. [37] models retweet network and posts by a community level diffusion model. Zhang et al. [38] models retweeting process with a hierarchical community-level information diffusion model, which combines semantic analysis and social influence analysis. Liu et al. [39] utilizes the heterogeneous link information and textual content to construct a generative graphical model and learn user topic interests and influence between users.

3 Preliminaries

Before we describe our method, we first briefly review our notation and terminology and introduce the task to be addressed.

3.1 Notation and terminology

Let directed graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ be the friend relationship network, where $\mathcal{U}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of directed links. Let $\mathcal{L}$ be the set of *purchase logs* which are represented as triples $\{(User, Item, Time)\}$, i.e., (u, i, t), indicating that user u adopted item i at time t. We assume that each item $i \in \mathcal{I}$ is associated with a mixture of words as its binary features, and let $\boldsymbol{A} \in \{0, 1\}^{M \times F}$ be item attribute matrix with $\boldsymbol{a}_i \in \{0, 1\}^F$ being the attribute vector of i, and M, F being the number of items and attributes respectively. e.g. An item (an restaurant) includes 10 attributes, i.e., Mexican food, Chinese food, French food, Italian food, RestaurantsTakeOut, BusinessParking-garage, BusinessParking-street, Business Parking-validated, BusinessParking-lot, BusinessParking-valet, and $\boldsymbol{a}_i\{1, 0, 0, 0, 1, 0, 1, 0, 0, 0\}$ is the attribute vector of item i.

For ease of reference, we list the basic notations in Table 1, and briefly introduce some basic concepts that we use in the remainder.

Table 1 Basic notations used in this paper

Symbol	Description
$\mathcal{S}, K$	Seed set, size of seed set
C, Z, M, F	Number of communities, topics, items and attributes
$\mathcal{I}, \mathcal{U}, \mathcal{E}, \mathcal{D}$	Set of items, users, links, potential-influence logs
U, E, D	Number of users, links, potential-influence logs
$\mathcal{E}_v, \mathcal{D}_v$	The set of links towards v, the set of potential-influence logs with v as a propagation target
E_v, D_v	Size of $\mathcal{E}_v$, $\mathcal{D}_v$
d	Potential-influence log
$\boldsymbol{\psi}_z$	Multinomial distribution over attribute values in matrix $\boldsymbol{A}$ specific to topic z
$\boldsymbol{\phi}_i$	Multinomial distribution over topics of item i
z_i	The topic of item i
w_i	The bag of attribute values of item i
$\boldsymbol{\pi}_v$	Multinomial distribution over community specific to user v
c_d, c'_d	Communities associated with user v and u of potential-influence log $d = (u, v, i)$
s_e, s'_e	Communities associated with user v and u of friend relation link $e = (u, v)$
$\boldsymbol{\theta}_c$	The topic interest of community c
$\eta_{c'c}$	Community-level influence strength of community c' and c
I_e	Indicator of the existence of link e
I_d	Indicator of the existence of potential-influence log d
ε	i.e. $(\varepsilon_0, \varepsilon_1)$, Beta priors to $\boldsymbol{\eta}$
$\rho, \alpha, \beta, \omega$	Dirichlet priors to $\boldsymbol{\pi}_v, \boldsymbol{\theta}_c, \boldsymbol{\psi}, \boldsymbol{\phi}$

Definition 1 (**Potential-influence log**) A potential-influence log $d = (u, v, i)$ is constructed from two purchase logs $\left(u, i, t_p\right)$ and $\left(v, i, t_q\right)$ where $t_q - t_p \leq \Delta$ and $(u, v) \in \mathcal{E}$, meaning that v purchasing item i is potentially influenced by user u. In practice, threshold Δ is set manually [35]. The set of potential-influence logs is denoted as $\mathcal{D}$.

Definition 2 (**Community**) A community $c \in [1, 2, ..., C]$ consists of nodes which share some common properties or preferences. Each user $v \in \mathcal{U}$ may belong to multiple communities with a multinomial distribution π_v. Accordingly, π_{vc} indicates the probability that user v be the member of community c.

Definition 3 (**Topic**) A topic $z \in [1, 2 \ldots Z]$ is a multinomial distribution over attributes of items, denoted as $\boldsymbol{\psi}_z$. Every potential-influence log d is associated with a topic z_i which is sampled for item i according to its topic distribution.

Definition 4 (**Community-topic relevance**) Every community c has a topic-relevant distributions, i.e., topic interest θ_{cz}, which represents that as the propagation target, c influences the purchase of an item on topic z.

Definition 5 (**Community-level influence**) We assume that each potential-influence log d=(u, v, i) depends on communities c', c of u, v according to the community-topic relevance θ_{cz} and the community-level influence strength $\eta_{c'c}$.

3.2 Task formulation

We formally define our task of community-based influence maximization as follows:

Given size of seed set K, a network $\mathcal{G}$, the set of potential-influence logs $\mathcal{D}$, the item attribute matrix $\boldsymbol{A}$, our task is to (1) first infer community membership distribution of user π_{vc}, latent community-topic relevance θ_{cz} and item-topic relevance $p(z \mid i)$ while capturing the community-level influence according to the potential-influence logs; (2) then compute the most influential nodes set with the size of K through an effective influence maximization algorithm.

According to the task, we need to solve two problems:

(1) designing a model Ξ to infer latent variables:

$$\mathcal{G}, \mathcal{D}, \boldsymbol{A} \xrightarrow{\Xi} \theta_{cz}, p(z \mid i), \pi_{vc}$$

(2) running a community-based influence maximization algorithm $\mathcal{F}$ that satisfies:

$$\theta_{cz}, p(z \mid i), \pi_{vc} \xrightarrow{\mathcal{F}} \mathcal{S}$$

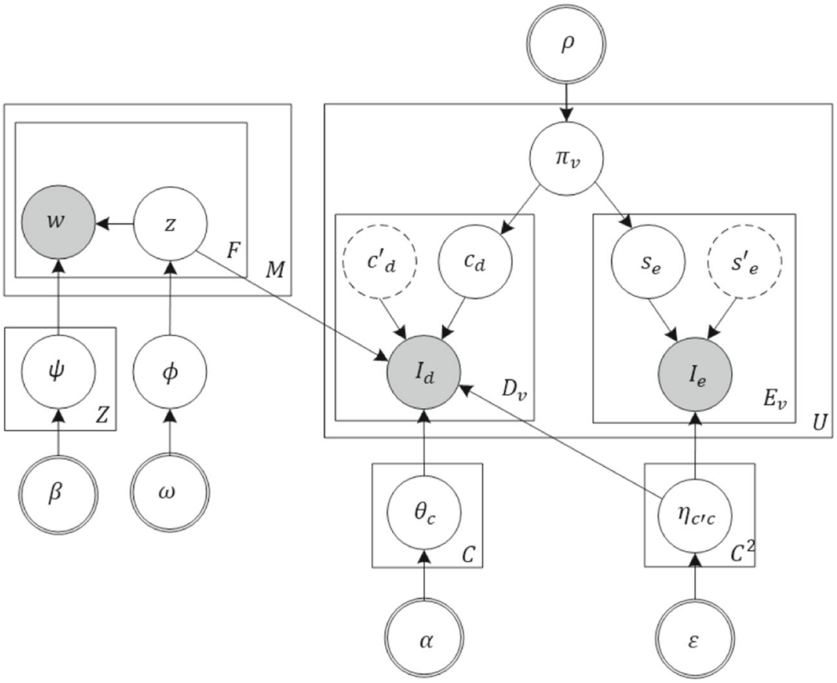
where $\mathcal{S}$ is the set of the most influential nodes, and $|\mathcal{S}| = K$.

4 The proposed method

To solve above problems, we propose the **CTIM**, a community-based influence maximization method that integrates a probabilistic generative model with a influence maximization algorithm to infer latent variables and select influential nodes effectively. The probabilistic generative model captures the community-topic relevance and item-topic relevance by jointly modeling influence diffusion and probabilistic topic, while the influence maximization algorithm identify the influential nodes with topic-aware and community-based strategy.

Our **CTIM** is composed of two steps. In Step 1, we utilize the *Comprehensive Latent Variable Model* to capture community-level topic interest $\boldsymbol{\theta}_c$, community membership distribution of each user and item-topic relevance. In Step 2, we infer community-to-community influence strength $\boldsymbol{P}(\boldsymbol{c} \mid \boldsymbol{z}, \boldsymbol{c'})$ using topic-irrelevant influence $\boldsymbol{\eta}_{\boldsymbol{c'c}}$ and community topic interest $\boldsymbol{\theta}_{\boldsymbol{c}}$, and further infer user-to-user influence strength using community-to-community influence strength and community membership distribution of each

Fig. 1 Graphical representation of our comprehensive latent variable model



user. Then we select the influential nodes using the thought of the divide and conquer method that presents superiority in efficiency. We will further detail the two steps in the following subsections.

4.1 Comprehensive latent variable model

4.1.1 Model description

Our comprehensive latent variable model is a probabilistic generative model of item attributes, user edges and their potential-influence logs inferred from the purchase logs. We take both influence logs and friend relation network into account when modeling influence diffusion for the observation that influence logs are sparser than friend relationship links. Thus, we use friend relation network as a supplementary to interact with potential-influence logs to mimic the process of influence diffusion. There are basically three components in our generative process, i.e., the generation of item attributes, user edges and potential-influence logs. The probabilistic graphical description of our generative model is presented in Fig. 1, and its generative process is presented in Algorithm 1.

To generate attributes of an item $i \in \mathcal{I}$, we draw a multinomial ϕ_i from a Dirichlet prior ω; then for each attribute w_{ij} in item i: (1) first draw a topic z_i from multinomial ϕ_i; (2) then draw an attribute w_{ij} from multinomial ψ_{z_i}.

For generating the user edge, we assume that each user may behave as the member of different communities in different settings. Thus, we denote each user to belong to different communities with a multinomial distribution $\boldsymbol{\pi}_v$, and use π_{vc} to denote the probability that user v acts as a member of community c. For each link $e\,(u, v) \in E$, user u acts as member of community s'_e, and user v acts as member of community s_e. A Bernoulli distribution $\eta_{s'_e s_e}$ is used to denote influence strength between community s'_e and community s_e, which determines the probability of the presence of link e.

When generating the potential-influence logs, for each potential-influence log $d = (u, v, i) \in D$, user u acts as member of community c'_d, and user v acts as member of community c_d. We define the topic interest of community c as $\boldsymbol{\theta}_c$. A Bernoulli distribution $\eta_{c'_d c_d}$ is used to denote influence strength between community c'_d and community c_d. The presence and absence of influence d between community c'_d and community c_d is denoted with a indicator I_d, which is determined by two factors: $\eta_{c'_d c_d}$ and $\boldsymbol{\theta}_c$. As in [37], the priors to $\eta_{c'c}$ is set as a $Beta(\varepsilon_0, \varepsilon_1)$, in the hyperparameters of which, negative samples are implicitly modeled, i.e., $\varepsilon_0 = \zeta ln(N_{neg}/C^2)$ and $\varepsilon_1 = 0.1$, where $N_{neg} = U(U-1)(1 + D/E) - D - E$ and ζ is a tunable weight.

Algorithm 1 GenerativeProcess

```
for each topic $z = 1, 2, \dots Z$ do
    Draw $\boldsymbol{\psi}_z \sim Dir(\beta)$;
end for
for each item $i = 1, 2, \dots M$ do
    Draw $z_i \sim Mul(\boldsymbol{\phi}_i)$;
    Draw $w_i \sim Mul(\boldsymbol{\psi}_{z_i})$;
end for
for each community $c = 1, 2, \dots C$ do
    Draw topic interest distribution $\theta_c \sim Dir(\alpha)$;
    for each community $c' = 1, 2, \dots C$ do
        Draw the community-level diffusion probability $\eta_{c'c} \sim Beta(\varepsilon_0, \varepsilon_1)$;
    end for
end for
for each user $v = 1, 2, \dots U$ do
    Draw the distribution over communities $\boldsymbol{\pi}_u \sim Dir(\rho)$;
    for each link $e = (u, v) \in \mathcal{E}_v$ do
        Draw user $u$'s community $s'_e \sim Mul(\boldsymbol{\pi}_u)$;
        Draw user $v$'s community $s_e \sim Mul(\boldsymbol{\pi}_v)$;
        Draw the existence indicator $I_e \sim Ber(\eta_{s'_e s_e})$;
    end for
    for each potential-influence log $d = (u, v, i) \in \mathcal{D}_v$ do
        Draw user $u$'s community $c'_d \sim Mul(\boldsymbol{\pi}_u)$;
        Draw user $v$'s community $c_d \sim Mul(\boldsymbol{\pi}_v)$;
        Draw item $i$'s topic indicator $z_i \sim Mul(\boldsymbol{\phi}_i)$;
        Draw the existence indicator $I_d \sim Ber(\eta_{c'_d c_d}\theta_{c_d z_i})$;
    end for
end for
```

4.1.2 Model inference

Following [40], we use collapsed Gibbs sampling method to estimate the parameters of our model. As item-topic generation is independent of influence diffusion in our model, our Gibbs sampling process is composed of two steps. First, we iteratively sample hidden variables $\boldsymbol{z}$. After the Gibbs Sampling converges, we use those samples to estimate the unknown parameters $\boldsymbol{\phi}, \boldsymbol{\psi}$ and obtain topic distribution z_i of each item i; Second, we iteratively sample hidden variables $\boldsymbol{s}, \boldsymbol{s'}, \boldsymbol{c}, \boldsymbol{c'}$. After the Gibbs Sampling converges, we estimate the unknown parameters $\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\eta}$ with the samples. Due to space constraints, we show only the derived Gibbs sampling formulas, omitting the detailed derivation process.

Sample latent topic z_{ij} for each attribute w_{ij} in matrix A

$$P(z_{ij} = z \mid \boldsymbol{z_{\neg ij}}, \boldsymbol{w}) \propto \frac{n_{i,z} + \omega}{\sum_Z (n_{i,z} + \omega)} \cdot \frac{n_{z,w_{ij}} + \beta}{\sum_{M \times F} (n_{z,w_{ij}} + \beta)} \tag{1}$$

where $n_{i,z}$ refers to the number of times that topic z has been observed with an attribute in item i, and $n_{z,w_{ij}}$ denotes the number of times that attribute w_{ij} has been observed with topic z.

After the Gibbs Sampling of latent topic converges, we can estimate parameters $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ as follows.

$$\phi_{iz} = \frac{n_{i,z} + \omega}{\sum_Z (n_{i,z} + \omega)} \tag{2}$$

$$\psi_{zw} = \frac{n_{z,w} + \beta}{\sum_{M \times F} (n_{z,w} + \beta)} \tag{3}$$

Also, we can infer item-topic relevance $P(z \mid i)$ by counts.

Sample community indicator s_e, s'_e for each edge $e = (u, v) \in \mathcal{E}$

$$P(s_e = c, s'_e = c' \mid \boldsymbol{s_{\neg e}}, \boldsymbol{s'_{\neg e}}, \cdot) \propto \frac{n_{u,c'} + \rho}{\sum_C (n_{u,c'} + \rho)} \cdot \frac{n_{v,c} + \rho}{\sum_C (n_{v,c} + \rho)} \cdot \frac{n_{c',c} + \varepsilon_1}{n_{c',c} + \varepsilon_{0+}\varepsilon_1} \tag{4}$$

where $n_{u,c'}$ indicates the number of times when user u acts as a member of community c' in all links and potential-influence logs, $n_{c',c}$ is the number of links from community c' to community c and potential-influence logs with community c' as source community and community c as target community.

Sample community indicator c_d, c'_d for each potential-influence log $d = (u, v, i) \in \mathcal{D}$ For each potential-influence log $d\,(u, v, i)$, we first draw item i's topic distribution $\boldsymbol{z_i}$ and then apply the following Gibbs sampling formula:

$$P(c_d = c, c'_d = c' \mid \boldsymbol{c_{\neg d}}, \boldsymbol{c'_{\neg d}}, \boldsymbol{z_{\neg i}}, z_i = z, \cdot) \propto \frac{n_{u,c'} + \rho}{\sum_C (n_{u,c'} + \rho)} \cdot \frac{n_{v,c} + \rho}{\sum_C (n_{v,c} + \rho)} \cdot \frac{n_{c',c} + \varepsilon_1}{n_{c',c} + \varepsilon_{0+}\varepsilon_1} \cdot \frac{n_{c,z} + \alpha}{\sum_Z (n_{c,z} + \alpha)} \tag{5}$$

where $n_{c,z}$ denotes the number of potential-influence logs which is relevant to topic z and with community c as propagation target.

Every potential-influence log d is associated with a topic z_i according to the item i in $d\,(u, v, i)$. Thus, according to item-topic relevance $P(z \mid i)$, $n_{c,z}$ can be computed as $\sum_M n_{c,i} \cdot P(z \mid i)$, where $n_{c,i}$ is the number of potential-influence logs which is about item i and with community c as propagation target. Thus replacing $n_{c,z}$ in Formula (5) with $\sum_M n_{c,i} \cdot P(z \mid i)$, we can get Formula (6) as follows.

$$P(c_d = c, c'_d = c' \mid \boldsymbol{c_{\neg d}}, \boldsymbol{c'_{\neg d}}, \boldsymbol{z_{\neg i}}, z_i = z, \cdot) \propto \frac{n_{u,c'} + \rho}{\sum_C (n_{u,c'} + \rho)} \cdot \frac{n_{v,c} + \rho}{\sum_C (n_{v,c} + \rho)} \cdot \frac{n_{c,c'} + \varepsilon_1}{n_{c,c'} + \varepsilon_{0+}\varepsilon_1} \cdot \frac{\sum_M n_{c,i} \cdot P(z \mid i) + \alpha}{\sum_Z (\sum_M n_{c,i} \cdot P(z \mid i) + \alpha)} \tag{6}$$

After all Gibbs Sampling converges, the unknown parameters can be estimated as follows.

$$\pi_{vc} = \frac{n_{v,c} + \rho}{\sum_C (n_{v,c} + \rho)} \tag{7}$$

$$\eta_{c'c} = \frac{n_{c',c} + \varepsilon_1}{n_{c',c} + \varepsilon_{0+}\varepsilon_1} \tag{8}$$

$$\theta_{cz} = \frac{\sum_M n_{c,i} \cdot P(z \mid i) + \alpha}{\sum_Z (\sum_M n_{c,i} \cdot P(z \mid i) + \alpha)} \tag{9}$$

4.2 Community-based influence maximization

In this section, we show how to infer influence strength between individual users specific to item i, introduce our influence computation model, and then present our community-based influence maximization algorithm.

4.2.1 Inference of user-to-user influence strength

The diffusion probabilities between two communities for all the topics $z \in [1, 2...Z]$ are composed of the production of community-level influence strength and propagation-target community's topic interest:

$$P(c \mid z, c') = \eta_{c'c} \cdot \theta_{cz}. \tag{10}$$

The user-to-user influence probabilities for all the topics $z \in [1, 2...Z]$ are determined by community memberships and community-level diffusion probabilities:

$$P(v \mid z, u) = \sum_{c,c'} \pi_{vc} \cdot \pi_{uc'} \cdot P(c \mid z, c'). \tag{11}$$

The user-to-user influence strength over item i is the weighted average of topic-relevant influence probabilities w.r.t. the topic multinomial distribution of item i:

$$P(v \mid i, u) = \sum_{z=1}^{Z} P(z \mid i) \cdot P(v \mid z, u). \tag{12}$$

4.2.2 Influence computation model

We utilize MIA model [19] as influence computation model in our algorithm since MIA has better performance and scalability, especially for large-scale social networks. MIA model proposes to estimate the influence from node u to node v using maximum influence path (MIP). For a given node v in the graph G, there are several paths from u to v which are non-cyclic sequences of users. We denote the set of paths from u to v as $Path(\mathcal{G}, u, v)$ and represent one path in the set as $P = \langle u = w_1, w_2, ...w_r = v\rangle$. In the independent cascade (IC) diffusion model, the probability of u activating v through the path P is the product of all probabilities of w_k activating w_{k+1}, where w_k is the node on the path P:

$$pp(P) = \prod_{k=1}^{r-1} pp(w_k, w_{k+1}). \tag{13}$$

Maximum influence path from u to v is defined as

$$MIP(u, v) = \arg\max_{P} \{pp(P) \mid P \in Path(\mathcal{G}, u, v)\}. \tag{14}$$

We build a maximum influence in-arborescence $MIIA$ (v, h) by assembling all the maximum influence paths from $u \in \mathcal{U} \setminus \{v\}$, with v as its root and h as threshold:

$$MIIA(v, h) = \cup_{u \in \mathcal{U}, pp(MIP(u,v)) \geq h} MIP(u, v). \tag{15}$$

The maximization influence out-arborescence $MIOA$ (v, h) is:

$$MIOA(v, h) = \cup_{u \in \mathcal{U}, pp(MIP(v,u)) \geq h} MIP(v, u). \tag{16}$$

If the activated probability of maximum influence path from u to v is less than h, i.e., $pp(MIP(u, v)) < h$, MIA model assumes v can not be activated by u. In this paper, we set $h = 0.1$. With $MIIA(v, h)$, the activated probability of node v is computed as follows.

$$ap(v \mid \mathcal{S}) = \begin{cases} 1, & v \in \mathcal{S} \\ 0, & N^{in}(v) = \varnothing \\ 1 - \prod_{w \in N^{in}(v)} (1 - ap(w \mid \mathcal{S}) \cdot pp(w, v)) & otheres \end{cases} \tag{17}$$

where $N^{in}(v)$ represents in-neighbors set of node v and $pp(w, v)$ denotes the probability of w activating v. Note that activated probability of node w, i.e., $ap(w \mid \mathcal{S})$, can be computed recursively through w's in-neighbors. Similarly, we can estimate the activated probabilities of all nodes and obtain the influence spread $I(\mathcal{S})$ of seed set $\mathcal{S}$:

$$I(\mathcal{S}) = \sum_{v \in V} ap(v \mid \mathcal{S}). \tag{18}$$

4.2.3 Community-based influence maximization algorithm

Divide-and-conquer is a problem-solving strategy that can predigest the scope of questions and reduce the complexity of computing. Motivated by this thought, we design a influence maximization algorithm based on community partition. Supposing we have inferred that all the users belong to corresponding community, i.e., community detection has been implemented, the main idea of our algorithm is to choose the community in which the k-th seed node comes into being, mine the k-th seed node in this exact community and repeat step 1 and step 2 K times until finding a seed set with K nodes. As CGA Algorithm in [22], we propose to use dynamic programming to choose which community the k-th seed node should come from. A maximum influence arborescence (MIA) model [19] is utilized as approximation for better performance in computing influence diffusion. Additionally, we accomplish community detection by inferring which community a user belongs to according to the values of community memberships of the user, i.e., the user belongs to the community corresponding to the largest community membership as shown in formula (19):

$$c_m^v \leftarrow \arg\max_{c} \pi_{v,c} \tag{19}$$

where $m \in [1, 2...C]$ and c_m^v denotes user v belongs to community c_m. When a user has two or more equal community memberships, we assign the user to an arbitrary community. Now we have accomplished the community detection since the communities that every node belongs to have been obtained.

Our work is different from the CGA Algorithm primarily in that (1) CGA assumes influence strength between pairs of nodes are known that decreases the difficulties of the algorithm, while our method computes influence strength between users by a comprehensive latent variables model and an inference process; (2) CGA adopts a two-step community detection algorithm to detect communities, the time complexity of which is non-trivial, while our method computes community memberships of users through latent variables generative model and Gibbs sampling that increases the efficiency largely; (3) after selecting the community, from which the k-th seed should generate, CGA mines the k-th seed using MixedGreedy algorithm, while our method employs MIA model as influence computation model, which is more efficient and scalable than MixedGreedy; (4) CGA is not topic-aware, while our method is topic-aware, that means the influence between pairs of users is measured by topic interest of users and different items with different topic distribution should be assigned different seed set. This can definitely improve prediction accuracy significantly. Our method is outlined as Algorithm 2.

Algorithm 2 Community-based influence maximization algorithm

```
Input: Directed graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, set of potential-
    influence logs $\mathcal{D}$, $\boldsymbol{\pi}$, $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, item-topic relevance $P(z \mid i)$,
    size of seed nodes $K$
Output: Seed nodes set $\mathcal{S}$.
 1: for $c = 1 \ldots C$ do
 2:     for $c' = 1 \ldots C$ do
 3:         for $z = 1 \ldots Z$ do
 4:             $P(c \mid z, c') = \eta_{c'c} \cdot \theta_{c,z}$;
 5:         end for
 6:     end for
 7: end for
 8: for $d = (u, v, i) \in \mathcal{D}$ do
 9:     for $c = 1 \ldots C$ do
10:         for $c' = 1 \ldots C$ do
11:             $P(v \mid z, u) = \sum_{c,c'} \pi_{vc} \cdot \pi_{uc'} \cdot P(c \mid z, c')$;
12:         end for
13:     end for
14: end for
15: for $i = 1 \ldots M$ do
16:     for $d = (u, v, i) \in \mathcal{D}$ do
17:         for $z = 1 \ldots Z$ do
18:             $P(v \mid i, u) = \sum_{z=1}^{Z} P(z \mid i) \cdot P(v \mid z, u)$;
19:         end for
20:     end for
21: end for
22: for $v = 1 \ldots U$ do
23:     $c_m^v \leftarrow \arg\max_c \pi_{v,c}$;
24: end for
25: $\mathcal{S} = \mathcal{S}_1 = \mathcal{S}_2 = \ldots = \mathcal{S}_C = \varnothing$
26: for $k = 1 \ldots K$ do
27:     $I[0, k] = 0$; $s[0, k] = 0$;
28: end for
29: for $m = 1 \ldots C$ do
30:     $I[m, 0] = 0$;
31: end for
32: for $k = 1 \ldots K$ do
33:     for $m = 1 \ldots C$ do
34:         $\Delta I_m = \max(I_m(\mathcal{S} \cup u) - I_m(\mathcal{S}))$, $u \in c_m$;
35:         $I[m, k] = \max(I([m-1, k], I([C, k-1] + \Delta I_m)$;
36:         if $I([C, k-1] + \Delta I_m \geq I([m-1, k]$ then
37:             $s[m, k] = m$;
38:         else
39:             $s[m, k] = s[m-1, k]$;
40:         end if
41:     end for
42:     $j \leftarrow s[C, k]$;
43:     $u_k \leftarrow \arg\max_{u \in c_j}(I(S_j \cup u) - I(S_j))$;
44:     $\mathcal{S}_j = \mathcal{S}_j \cup u_k$; $\mathcal{S} = \mathcal{S} \cup u_k$
45: end for
```

In lines 1–21, Algorithm 2 first computes user-to-user influence strength on all items. In lines 22–24, the algorithm compares the community membership values and accomplishes community detection. In lines 25–31, the algorithm does some initializations. In lines 32–41, with MIA model as influence computation model, the algorithm computes which community the k-th seed node should come from. It computes marginal influence of any node u in community m under seed set $\mathcal{S}$ and selects the maximum marginal influence as marginal influence w.r.t. community m under seed set $\mathcal{S}$. It applies dynamic programming process to select community c_j, from which the k-th seed node should be mined. Then, in lines 42–45, the algorithm utilizes MIA model to find the exact k-th seed node in community c_j by the rule of maximum marginal influence, and updates the targeted seed set $\mathcal{S}$ as well as seed set $\mathcal{S}_j$ for the community. After K iterations, the algorithm finds the target seed set $\mathcal{S}$ with size of K.

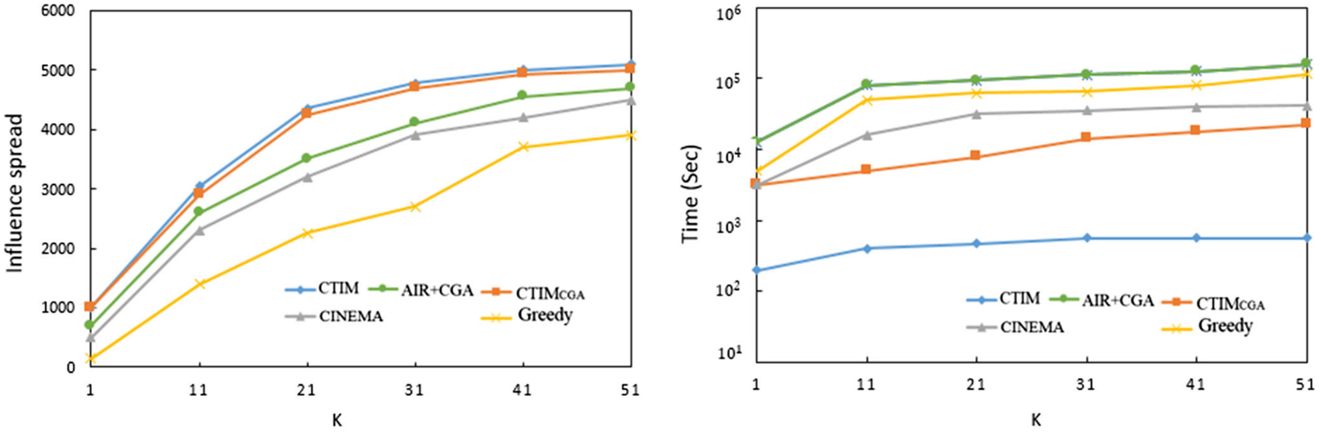
4.3 Complexity analysis

We proceed to analyze the time complexity of our **CTIM** method. First, we discuss the time complexity of the generative process. Given parameters C and Z fixed, the time of the generative process is linear to the size of input data and times of iterations, i.e., $O(T_1 M + T_2(E + D))$, where T_1 denotes times of iterations in sampling latent topics and T_2 denotes times of iterations in sampling $\boldsymbol{s}, \boldsymbol{s'}, \boldsymbol{c}, \boldsymbol{c'}$. In each iteration in sampling latent topics, the time complexity is $O(ZF)$. In each iteration in sampling $\boldsymbol{s}, \boldsymbol{s'}, \boldsymbol{c}, \boldsymbol{c'}$, the time complexity of sampling community indicators associated with each link and each potential-influence log is $O(C^2)$.

Second, we discuss the complexity of our community-based influence maximization algorithm (Algorithm 2). It takes $O(C^2 Z + C^2 D + MDZ + UC)$ time to compute user-to-user influence strength on all items and compare community membership values (lines 1–24). It takes $O(K)$ time to initialize the arrays $I[0, k]$ and $s[0, k]$, and takes $O(C)$ time to initialize $\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_C$ and the array $I[m, 0]$ (lines 25–31). Supposing the largest community is

Table 2 Feature comparison of different methods

	Features	
	Community-based	Topic-aware
Greedy		
CINEMA	•	
AIR+CGA	•	•
CTIM$_{CGA}$	•	•
CTIM	•	•



(a) Influence spread

(b) Running time

Fig. 2 Performance comparison experiments on Yelp dataset

c_p, lines 32–41 need $O(KCT_p)$ time where T_p is the time to compute the influence spread of a node in community c_p. Suppose the maximal running time expended to compute $MIIA(v, \theta)$ is t_{ip} for any $v \in c_p$, thus the running time of T_p is $O(\mid c_p \mid t_{ip} + n_{ip}n_{op} \log \mid c_p \mid)$, where $\mid c_p \mid$ denotes the number of nodes in community c_p, $n_{ip} = \max_{v \in c_p}\{\mid MIIA(v, \theta) \mid\}$ and $n_{op} = \max_{v \in c_p}\{\mid MIOA(v, \theta) \mid\}$. It takes $O(K \mid c_p \mid T_p)$ time for lines 42–45 to mine nodes using MIA model. Therefore, the time complexity of algorithm 3 is $O(KCT_p + K \mid c_p \mid T_p)$ if $O(KCT_p + K \mid c_p \mid T_p) >> O(C^2Z + C^2D + MDZ + UC)$.

5 Experimental evaluation

5.1 Data set

We use two real-world and publicly available datasets: Yelp dataset challenge 2014[1] and Digg dataset.[2] Yelp dataset challenge 2014 contains 366,715 users, 2,949,285 links, 61,184 items, and Digg dataset contains 30,358 users, 99,846 directed arcs, 7,100 items.

For Yelp dataset challenge 2014, we do not have data directly reflecting purchase time, so we assume the time at which the user reviews the item is the time when the item is purchased. Besides, we disregard the unfrequent behaviors of repeated review for the same item.

5.2 Baselines and parameters

We compare our CTIM with several state-of-the-art baselines. Table 2 lists the features of all methods.

Greedy: This is the original greedy algorithm [3].

CINEMA: Proposed in [24], it is a community-based influence maximization algorithm.

AIR+CGA: As CGA does not consider topics, we extend CGA to AIR+CGA to support our comparison experiments. This is a method that integrates topic diffusion model AIR [35] with community detection and seed-set selection of CGA.

CTIM$_{CGA}$: As CGA does not consider topics, we extend CGA to CTIM$_{CGA}$ to support our comparison experiments. This is a variation of CTIM by swapping out our seed-set selection procedure with seed-set selection of CGA. The only difference between CTIM$_{CGA}$ and CTIM is influence computation model.

We use 60% potential influence logs as train set and 20% potential influence logs as validation set, and set 20% potential influence logs and all links in friend relationship graph as test set. We train all methods by setting the size of seed set K to vary from 1 to 50. We train CTIM and the baseline CTIM$_{CGA}$ by setting the number of topics Z to vary from 2 to 16 and setting the number of communities C to vary from 25 to 150.

As regards to the hyperparameters $\{\alpha, \rho, \beta, \omega, \varepsilon\}$, we adopt a fixed value, i.e., $\rho = 50/C, \beta = 0.01, \alpha = 50/Z, \omega = 50/Z$, and $\varepsilon_0, \varepsilon_1$ are set as Section 4.

5.3 Results

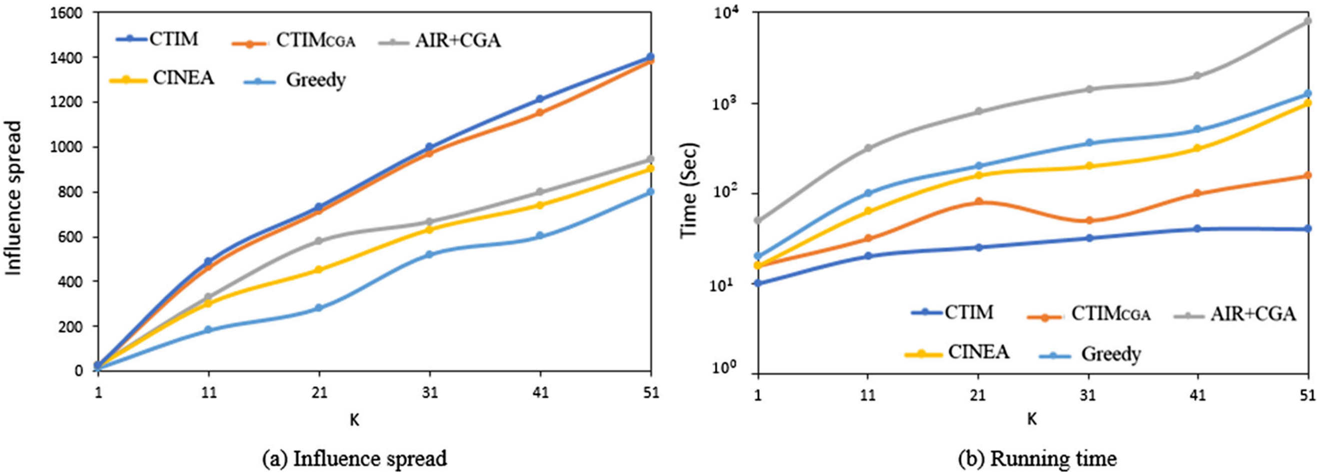
For evaluating performance of all methods, we use two metrics, influence spread and running time.

We compare the performance of CTIM with four baselines for two datasets; we analyze the effect of parameters Z (number of topics) and C (number of communities) on the performance of our CTIM for Yelp dataset.

Performance comparison The methods of performance comparison are described as follows. First, for fairness,

[1] https://www.yelp.com/

[2] https://www.digg.com/



(a) Influence spread

(b) Running time

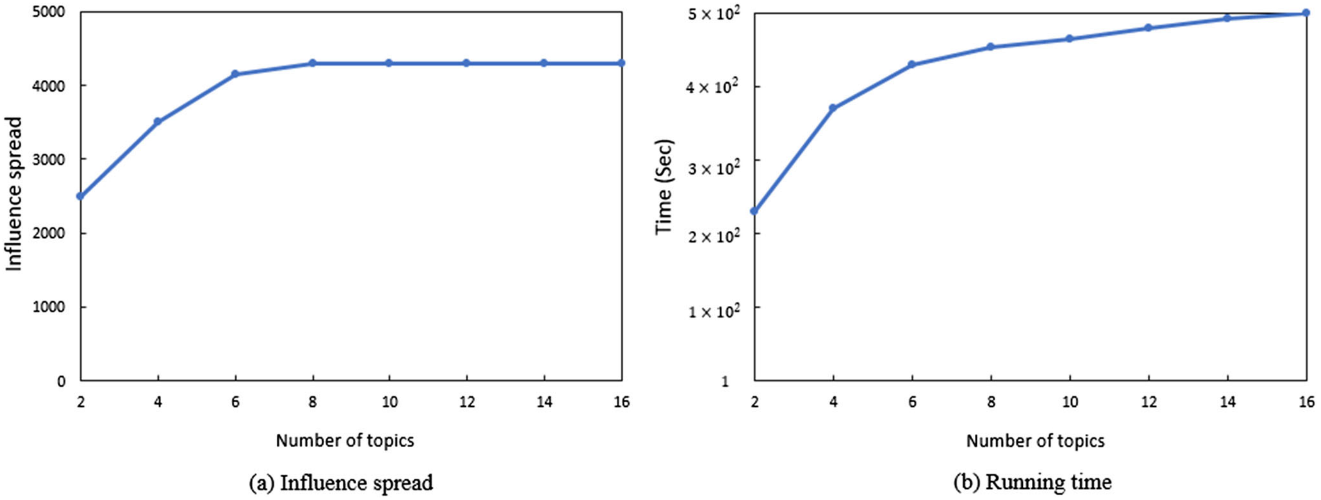
Fig. 3 Performance comparison experiments on Digg dataset

we fix parameters $Z = 8$ and $C = 100$ for CTIM and CTIM$_{CGA}$, and fix parameter $C = 100$ for AIR+CGA and CINEMA according the features of the algorithms (see Table 2). We do not set parameters Z and C for Greedy since Greedy is topic-blind and community-blind. Second, we evaluate the performance of different algorithms on two datasets by varying the parameter K from 1 to 50. The results of performance comparison on Yelp dataset and Digg dataset are respectively shown in Figs. 2 and 3.

In Figs. 2a and 3a, we can see that CTIM significantly outperforms Greedy, CINEMA and AIR+CGA w.r.t. influence spread because we employ a more effective topic-aware method, i.e., our comprehensive latent variable model, which takes full advantage of semantic information, jointly applies influence diffusion modeling and probabilistic topic modeling, and improves the accuracy of prediction largely. As Figs. 2a and 3a shows, the curve of CTIM is almost overlapping with that of CTIM$_{CGA}$ and the reason is that these two methods are only different in influence computation model, which does affect influence spread hardly (consistent with the previous experiments in [19]).

As illustrated in Figs. 2a and 3a, the influence spread of all the algorithms with topic-aware strategies outperform that of algorithms with topic-blind strategy. This is because topic-aware strategy can improve the accuracy of prediction (verified by previous work [35]). It can also be observed from Figs. 2a and 3a that AIR+CGA outperforms Greedy and CINEMA but is inferior to CTIM$_{CGA}$ and CTIM. This is due to that although AIR+CGA is with topic-aware strategy, simply using Expectation-Maximization method makes it incapable of capturing semantic information and extracting topic distribution well.

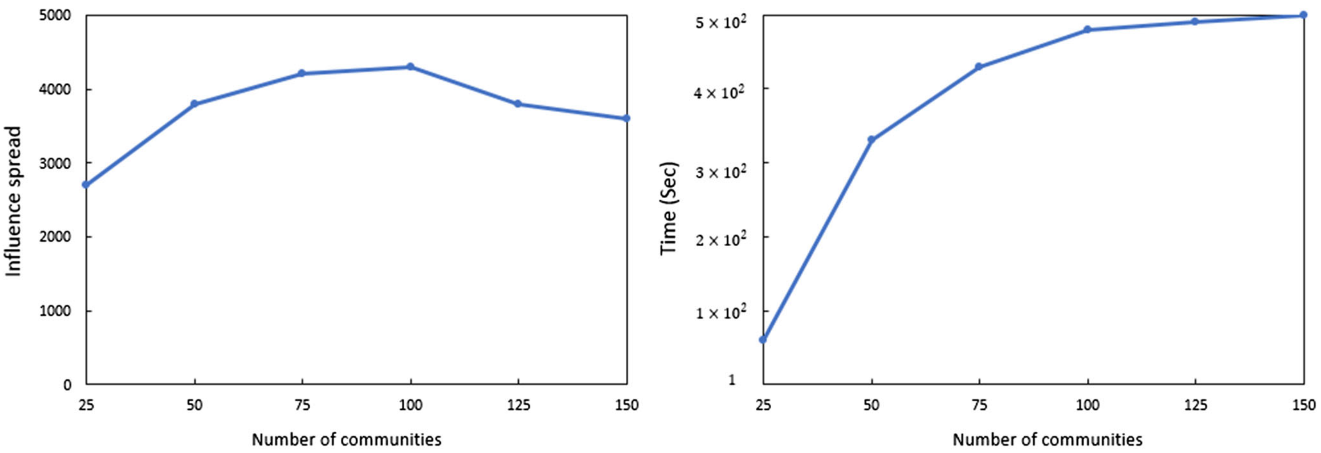
It is remarkable in Figs. 2b and 3b, that our CTIM is faster than all the baselines by orders of magnitude. The reason is that CTIM integrates the community detection into the influence diffusion modeling that improves the



(a) Influence spread

(b) Running time

Fig. 4 Impact of parameter Z on Yelp dataset



(a) Influence spread

(b) Running time

Fig. 5 Impact of parameter C on Yelp dataset

efficiency of the algorithm largely and our latent variables model is dramatically more efficient than AIR topic diffusion model. Specially, CTIM is much faster than CTIM$_{CGA}$ because CTIM is based on MIA influence computation model which is orders of magnitude more efficient than that of CTIM$_{CGA}$.

Impact of parameter *Z* We analyze the impact of number of topics on the performance of our CTIM algorithm by varying the parameter Z from 2 to 16. We set $K = 20$ and $C = 100$. The results are demonstrated in Fig. 4. As illustrated in Fig. 4a, when Z varies from 2 to 8, the influence spread improves significantly. And with the number of topics increasing, the influence spread of CTIM reaches a plateau. This is possibly because when making purchase decisions, users usually do not distinguish topics very punctiliously, thus, the overestimated topics may generate overfitting problems, without contributing to the increase of influence spread. The results in Fig. 4a exactly testify the robust and insensitiveness of our CTIM to the number of latent topics.

From Fig. 4b, it can be seen that with the increase of number of topics, the running time of CTIM increases relatively stably. This is due to the fact that only Algorithm 2 is affected by Z, that takes a relatively small percentage in the whole process of influence maximization problem resolution, thus, the increase of Z does not significantly affect the overall performance.

Impact of parameter *C* We evaluate the sensitiveness of CTIM algorithm to number of communities and the experiment results is as Fig. 5 shows. Note that we set $K = 20$ and $Z = 8$. Figure 5a presents when C varies from 25 to 100, the influence spread of CTIM improves significantly, reaches a peak with C increasing to 100, and then decreases evidently when C continues to grow. This is because when the number of communities increasing (below 100), users can be divided into finer-grained communities with more precise influence pattern accordingly, which yields better predicting performance. However, when the number of communities beyond 100, the error caused by the difference between influence spread computed within communities and influence spread computed over the whole network increases significantly that results in the reduction of performance.

From Fig. 5b, we observe that the running time of CTIM is more sensitive to number of communities than to number of topics. It is because both Algorithm 1 and Algorithm 2 are affected by C, that makes C be more important than Z in the whole process of influence maximization problem resolution.

6 Conclusions

In this paper, we propose a new influence maximization method, CTIM, which exploits topic-aware and community-based strategy to improve the performance of influence maximization. A comprehensive latent variable model is proposed that can capture item topic relevance and topic interest across communities, avoiding volatility of individual users and alleviating data-sparsity problem. Besides, the proposed model enables integrating the community detection into the influence diffusion modeling instead of performing community detection independently, which improves the efficiency of the algorithm largely. We design a collapsed Gibbs sampling algorithm to infer model parameters. We propose a community-based heuristic algorithm to mine influential nodes. We evaluate effectiveness and efficiency of our proposals by comprehensive experiments.

The experiment results validate our method and show the superiority of our algorithm compared with state-of-the-art influence maximization algorithms. As to future work, we plan to extend our CTIM algorithm under Linear Threshold Model. In addition, we plan to explore influence maximization problem w.r.t. dynamic evolution of social networks.

Acknowledgements This work is supported by National Key R & D Program of China Project #2017YFB0203201 and Australian Research Council Discovery Project DP150104871.

References

1. Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of SIGKDD, pp 57–66
2. Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of SIGKDD, pp 61–70
3. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proceedings of SIGKDD, pp 137–146
4. Galhotra S, Arora A, Roy S (2016) Holistic influence maximization: combining scalability and efficiency with opinion-aware models. In: Proceedings of SIGMOD, pp 743–758
5. Goyal A, Lu W, Lakshmanan LV (2011) CELF++: opti- mizing the greedy algorithm for influence maximization in social networks. In: Proceedings of WWW, pp 47–48
6. Kundu S, Murthy C, Pal SK (2011) A new centrality measure for influence maximization in social networks. In: Proceedings of international conference on pattern recognition and machine intelligence, pp 242–247
7. Leskovec J, Krause A, Guestrin C, Faloutsos C, Van Briesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: Proceedings of SIGKDD, pp 420–429
8. Cheng S, Shen H, Huang J (2012) StaticGreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of ACM international conference on information and knowledge management, pp 509–518
9. Tang Y, Shi Y, Xiao X (2015) Influence maximization in near-linear time: a martingale approach. In: Proceedings of SIGMOD, pp 1593–1554
10. Tang Y, Xiao X, Shi Y (2014) Influence maximization: near-optimal time complexity meets practical efficiency. In: Proceedings of SIGMOD, pp 75–86
11. Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proceedings of SIGKDD, pp 199–208
12. Wang Y, Feng X (2009) A potential-based node selection strategy for influence maximization in a social network. In: Proceedings of international conference on advanced data mining and applications, pp 350–361
13. Su J, Havens TC (2015) Quadratic program-based modularity maximization for fuzzy community detection in social networks. IEEE Trans Fuzzy Syst 23(5):1356–1371
14. Yang J, McAuley J, Leskovec J (2013) Community detection in networks with node attributes. In: Proceedings of ICDM, pp 1151–1156
15. Li J, Hu X, Jian L, Liu H (2017) Toward time-evolving feature selection on dynamic networks. In: Proceedings of ICDM, pp 1003–1008
16. Fang Y, Cheng R, Luo S, Hu J (2016) Effective community search for large attributed graphs. Proc VLDB Endow 9(12):1233–1244
17. Kalish S (1985) A new product adoption model with price, advertising, and uncertainty. Manag Sci 31(12):1569–1585
18. Luo ZL, Cai WD, Li YJ (2012) Recent progress in data engineering and internet technology. Springer, Berlin
19. Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of SIGKDD, pp 1029–1038
20. Cao T, Wu X, Wang S, Hu X (2010) Oasnet: an optimal allocation approach to influence maximization in modular social networks. In: Proceedings of ACM symposium on applied computing, pp 1088–1094
21. Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):1–6
22. Wang Y, Cong G, Song G, Xie K (2010) Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In: Proceedings of SIGKDD, pp 1039–1048
23. Chen YC, Zhu WY, Lee WC, Lee S-Y (2014) Cim: community-based influence maximization in social networks. ACM Trans Intell Syst Technol 5(2):1–25
24. Li H, Bhowmick SS, Sun A, Cui J (2015) Conformity- aware influence maximization in online social networks. VLDB J 24(1):117–141
25. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res Arch 3:993–1022
26. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: Proceedings of international conference on World Wide Web, pp 1445–1456
27. Yin J, Wang J (2014) A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of SIGKDD, pp 233–242
28. Yang Z, Kotov A, Mohan A, Lu S (2015) Parametric and non-parametric user-aware sentiment topic models. In: Proceedings of SIGIR, pp 413–422
29. Iwata T, Yamada T, Sakurai Y, Ueda N (2010) Online multiscale dynamic topic models. In: Proceedings of SIGKDD, pp 663–672
30. Liang S, Yilmaz E, Kanoulas E (2016) Dynamic clustering of streaming short documents. In: Proceedings of SIGKDD, pp 995–1004
31. Zhao Y, Liang S, Ren Z, Ma J, Yilmaz E, Rijke MD (2016) Explainable user clustering in short text streams. In: Proceedings of SIGIR, pp 155–164
32. Cha Y, Cho J (2012) Social-network analysis using topic models. In: Proceedings of SIGIR, pp 565–574
33. Guo W, Wu S, Wang L, Tan T (2015) Social-relational topic model for social networks. In: Proceedings of CIKM, pp 1731–1734
34. Bi B, Tian Y, Balmin A, Balmin A, Cho J (2014) Scalable topic-specific influence analysis on microblogs. In: Proceedings of WSDM, pp 513–522
35. Barbieri N, Bonchi F, Manco G (2012) Topic-aware social influence propagation models. In: Proceedings of ICDM, pp 81–90
36. Chen W, Wei L, Zhang N (2012) Time-critical influence maximization in social networks with time-delayed diffusion process. In: Proceedings of AAAI, pp 592–598
37. Hu Z, Yao J, Cui B, Xing E (2015) Community level diffusion extraction. In: Proceedings of SIGMOD, pp 1555–1569
38. Zhang Y, Lyu T, Zhang Y (2017) Hierarchical community-level information diffusion modeling in social networks. In: Proceedings of SIGIR, pp 753–762
39. Liu L, Tang J, Han J, Jiang M, Yang S (2010) Mining topic-level influecne in heterogeneous networks. In: Proceedings of CIKM, Toronto, pp 199–208
40. Heinrich G (2008) Parameter estimation for text analysis. Technical Report, pp 1–32
41. Zhang Y, Chen H, Lu J, Zhang G (2017) Detecting and predicting the topic change of knowledge-based systems: a topic-based

bibliometric analysis from 1991 to 2016. Knowl.Based Syst 133:255–268

42. Della Rocca P, Senatore S, Loia V (2017) A semantic-grained perspective of latent knowledge modeling. Inf Fusion 36:52–67
43. Rizun N, Ossowska K, Taranenko Y (2018) Modeling the customer's contextual expectations based on latent semantic analysis algorithms. In: Światek J, Borzemski L, Wilimowska Z (eds) Information systems architecture and technology: proceedings of 38th international conference on information systems architecture and technology - ISAT 2017. ISAT 2017. Advances in intelligent systems and computing, vol 656. Springer, Cham
44. Wang Z, Gu S, Xu X (2018) GSLDA: LDA-based group spamming detection in product reviews. Appl Intell 48:3094