The VLDB Journal (2015) 24:117–141
DOI 10.1007/s00778-014-0366-x

REGULAR PAPER

Conformity-aware influence maximization in online social networks

Hui Li · Sourav S. Bhowmick · Aixin Sun · Jiangtao Cui

Received: 11 October 2013 / Revised: 30 June 2014 / Accepted: 11 July 2014 / Published online: 5 August 2014

Abstract Influence maximization (IM) is the problem of finding a small subset of nodes (seed nodes) in a social network that could maximize the spread of influence. Despite the progress achieved by state-of-the-art greedy IM techniques, they suffer from two key limitations. Firstly, they are inefficient as they can take days to find seeds in very large real-world networks. Secondly, although extensive research in social psychology suggests that humans will readily conform to the wishes or beliefs of others, surprisingly, existing IM techniques are *conformity-unaware*. That is, they *only* utilize an individual's ability to influence another but ignores *conformity* (a person's inclination to be influenced) of the individuals. In this paper, we propose a novel *conformity-aware cascade* (C^2) *model* which leverages on the interplay between influence and conformity in obtaining the influence probabilities of nodes from underlying data for estimating influence spreads. We also propose a variant of this model called C^3 *model* that supports *context-specific* influence and conformity of nodes. A salient feature of these models is that they are aligned to the popular *social forces principle* in social psychology. Based on these models, we propose a novel greedy algorithm called CINEMA that generates high-quality seed set for the IM problem. It first partitions, the network into a set of non-overlapping subnetworks and for each of these subnetworks it computes the *influence* and *conformity indices* of nodes by analyzing the sentiments expressed by individuals. Each subnetwork is then associated with a COG-*sublist* which stores the *marginal gains* of the nodes in the subnetwork in descending order. The node with maximum marginal gain in each COG-sublist is stored in a data structure called MAG-*list*. These structures are manipulated by CINEMA to efficiently find the seed set. A key feature of such partitioning-based strategy is that each node's influence computation and updates can be limited to the subnetwork it resides instead of the entire network. This paves way for seamless adoption of CINEMA on a distributed platform. Our empirical study with real-world social networks comprising of millions of nodes demonstrates that CINEMA as well as its context-aware and distributed variants generate superior quality seed set compared to state-of-the-art IM approaches.

Keywords Social networks · Influence maximization · Conformity · Network partitioning · Greedy algorithm

1 Introduction

Given a social network as well as an *influence propagation* (or *cascade*) model, the problem of *influence maximization* (IM) is to find a set of initial users of size k (referred to as *seeds*) so that they eventually influence the largest number of individuals (referred to as *influence spread*) in the network [23].

Electronic supplementary material The online version of this article (doi:10.1007/s00778-014-0366-x) contains supplementary material, which is available to authorized users.

Part of H. Li's work was done when he was pursuing doctoral degree in Nanyang Technological University.

H. Li · J. Cui
School of Computer Science and Technology, Xidian University,
Xi'an, China
e-mail: hli@xidian.edu.cn

J. Cui
e-mail: cuijt@xidian.edu.cn

S. S. Bhowmick (✉)· A. Sun
School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore
e-mail: assourav@ntu.edu.sg

A. Sun
e-mail: axsun@ntu.edu.sg

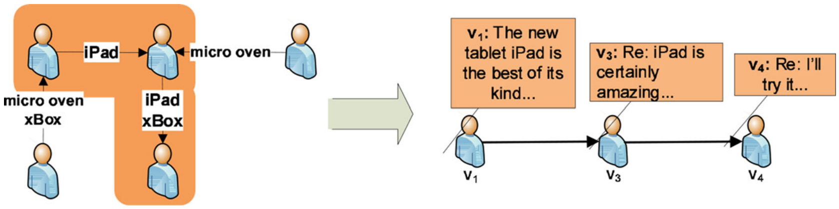

iPad
micro oven
micro oven
xBox
iPad
xBox
v1: The new tablet iPad is the best of its kind...
v3: Re: iPad is certainly amazing...
v4: Re: I'll try it...
v1
v3
v4

Fig. 1 An example of real-world social network

Pedro and Matt [34] and Domingos and Richardson [37] are the first to study influence maximization as an algorithmic problem. Kempe et al. [23] are the first to consider the problem of choosing the seeds as a discrete optimization problem. They proved that the optimization problem is NP-hard, and presented a greedy approximate algorithm applicable to three popular cascade models, namely the *independent cascade* (IC) model, the *weighted cascade* (WC) model, and the *linear threshold* (LT) model. A key strength of this algorithm is that it guarantees that the influence spread is within $(1 - 1/e)$ of the optimal influence spread where e is the base of the natural logarithm. However, deployment of these techniques on large-scale social networks is infeasible as they have poor efficiency and scalability [8]. Recently, several greedy approaches [8,28,42] were proposed to address this issue. While these approaches have been able to make significant progress in reducing the computation cost of the IM problem, they still suffer from the following limitations.

- Some of the aforementioned greedy approaches still take days to find seeds in real-world networks containing millions of nodes [17]. To alleviate this bottleneck, several heuristic-based techniques [6–9,18,24] have been proposed which are orders of magnitude faster than the greedy approaches. However, despite the blazing speed of these heuristic-based techniques, greedy approaches are more reliable as the former often produces inferior-quality seed set (detailed in Sect. 2). Note that seed set quality is paramount to companies as they would like to maximize the influence spreads of their new products.
- All these greedy and heuristic-based techniques assume that the influence probability of an edge $\overrightarrow{uv}$ depends *only* on node v's ability to influence u. Typically, this influence is determined by an independent probability (i.e., IC) or a probability proportional to the node degree (i.e., WC) or even a binary value controlled by a threshold (i.e., LT). Surprisingly, these techniques ignore the *conformity* of u, which refers to the inclination of u to be influenced by others (e.g., v) by yielding to perceived group pressure and copying the behavior and beliefs of others [2–4]. It is well known that humans will readily conform to the wishes or beliefs of others [2,4]. It was perhaps a surprise when Asch [3,4] found that people will do this even in cases where they can obviously determine that others are incorrect. Although the notion of conformity has been studied extensively in social psychology [3,4,10,14,19,39] and more recently in neuroscience [13,25], to the best of our knowledge, it has not been investigated in the context of online IM problem.

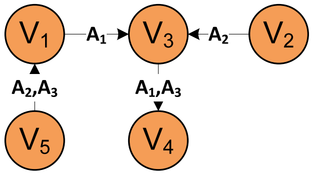


Fig. 2 Graph representation of Fig. 1

In this paper, we address the above limitations by proposing a novel greedy approach which is not only more efficient than state-of-the-art greedy techniques but it is also *conformity-aware*. That is, it exploits the interplay of influence and conformity of nodes in the underlying network to find high-quality seeds efficiently.

1.1 Why conformity matters?

Although conformity of human behavior is widely acknowledged by social psychologists, does it influence the IM problem? In this section, we motivate our work by answering this question affirmatively using an example. Consider Fig. 1 which depicts a fragment of a real-world social network consisting of five individuals. The label of an edge (e.g., "`iPad`") indicates the topic of conversation between the source and target individuals. To make it more discernible, part of the conversation is magnified in the right-hand side. An edge pointing from u to v ($\overrightarrow{uv}$) denotes the influence propagation path with respect to the topic labeled on the edge. We can represent this network using the graph depicted in Fig. 2 where each node denotes an individual.

Suppose a company wants to present a free trial version of an *iPad* to one of these individuals such that she is most likely to recommend her friends to buy an *iPad* in the future. That is, we aim to select a single seed node ($k = 1$) to propagate a piece of information (e.g., *iPad*). Let us review the seed selection in an existing *conformity-oblivious* greedy algorithm under IC model first. Assume that influence propagates within the network with probability $p = 0.5$. We need to calculate the expected influence size for all the nodes and select the highest one. Let X be the set of edges that are activated, through which influence propagates, and $\sigma^X(v)$ be the

Table 1 Expected influence size of nodes in Fig. 2

Model	$\sigma(v_1)$	$\sigma(v_2)$	$\sigma(v_3)$	$\sigma(v_4)$	$\sigma(v_5)$
IC	1.75	1.75	1.5	1	1.875
WC	1.67	1.67	2	1	1.83
C^2	1.66	1.66	1.04	1	1.06
C^3 (for A_1)	1.73	1	1.49	1	1

Table 2 Nodes' influence and conformity indices

Node ID	$\Phi(\cdot)$	$\Omega(\cdot)$	$\Phi_1(\cdot)$	$\Omega_1(\cdot)$
v_1	0.68	0.21	0.70	0.17
v_2	0.68	0.11	–	–
v_3	0.18	0.94	0.70	0.70
v_4	0.03	0.21	0.17	0.70
v_5	0.18	0.11	–	–

number of nodes that can be reached on activated edge paths from v. Thus, the expected number of influenced nodes from v [denoted as $\sigma(v)$] can be expressed as follows [23].

$$\sigma(v) = \sum_{X} Prob[X] \cdot \sigma^{X}(v) \tag{1}$$

In the above equation, $Prob[X]$ denotes the probability that all edges in X are activated. For instance, the expected influence size of v_3 under the IC model can be computed as $\sigma(v_3) = Prob[\overrightarrow{v_3v_4} \notin X] \times 1 + Prob[\overrightarrow{v_3v_4} \in X] \times 2$. As $Prob[\overrightarrow{v_3v_4} \notin X]$ or $Prob[\overrightarrow{v_3v_4} \in X]$ equals to 0.5, $\sigma(v_3)$ is 1.5. Table 1 reports the expected influence sizes of the five nodes under the IC and WC models (first two rows). Based on Table 1, we may select v_5 (resp. v_3) as the seed under the IC (resp. WC) model as it exhibits the highest expected influence size.

Unfortunately, this might not be the best choice when conformity of nodes are taken into account. Specifically, in real applications, the neighbors of a node (e.g., v_1 of v_5) may exhibit different conformity behavior. Observe that v_5 cannot influence anyone else unless $\overrightarrow{v_5v_1}$ is activated. The second and third columns in Table 2 report the *influence* [denoted by $\Phi(\cdot)$] and *conformity* [denoted by $\Omega(\cdot)$] *values* of all nodes, respectively. Intuitively, these values are computed by analyzing the sentiments expressed by edges in the underlying network (detailed in Sect. 3). Clearly, v_5 exhibits very small influence, whereas at the same time v_1 exhibits low conformity. Note that the lower the conformity of a node the less likely it is to be influenced by another. In other words, v_1 is not easily influenced by v_5. Consequently, in reality, $\overrightarrow{v_5v_1}$ is hardly activated during influence propagation! *Hence, state-of-the-art* IM *techniques may generate poor quality seed set as conformity of nodes are ignored during seed selection.*

1.2 Overview & contributions

In this paper, we present a novel greedy algorithm called CINEMA (**C**onformity-aware **IN**flu**E**nce **MA**ximization) to solve the IM problem in real-world social networks by effectively utilizing the interplay of conformity and influence. Pivotal to CINEMA is a *conformity-aware cascade model* (C^2) which provides a formal framework to obtain the influence probabilities by leveraging the influence and conformity of nodes. A *context-aware* variant of this model, namely C^3 model, is further proposed to exploit contextual information (whenever available) associated with these nodes toward this goal.

CINEMA first partitions the network into a set of non-overlapping *components* (subnetworks) and then *distribute* the *conformity-aware* influence maximization computation to these components. As shown in [15], many large real-world social networks are comprised of a set of clusters, each of which was defined as closely connected component, a piece of information can easily spread within the cluster (component) but hard to propagate from one to another. Hence, a node's influence in one component is not significantly affected by nodes in other components. Consequently, each node's influence computation and updates can be limited to the component it resides in. For each of these subnetworks, CINEMA computes the *influence* and *conformity indices* of nodes by leveraging an algorithm called CASINO [29] (detailed in Sect. 3).

Next, CINEMA selects the seed set S from the subnetworks. Specifically, a node v's selection into S is influenced by the conformity indices of the nodes around v at each iteration. A key challenge in this process is to determine the subnetworks from which the seeds need to be selected. To address this issue, inspired by [8,28], we present an efficient data structure called MAG-*list* (**MA**rginal **G**ain List), which stores the *candidate node* having maximum *marginal gain* from each component in the network and guides us to determine the members of seed set. MAG-list is space-efficient as it only requires $O(\ell)$ space complexity, where ℓ is the number of partitioned subnetworks. Thus, in contrast to majority of existing greedy approaches, we do not need to keep the entire collection of nodes of the network in the memory, which is prohibitively expensive for very large networks. Additionally, it provides an efficient framework to update the influence of nodes. Note that whenever a node is selected into the seed set, some other nodes' influence may change as well. Thus, it is important to dynamically update the influence of each node. Particularly, CINEMA applies an *on-demand update* strategy in each round to update the MAG-list. Only when a node in the MAG-list is selected as a potential candidate for the seed set, CINEMA updates all the nodes in the *component gain sublist* (COG-sublist) of this node (space complexity is $O(n')$ where n' is the maximum number of nodes in a subnetwork). It is not necessary to update all nodes in the MAG-list.

Observe that due to the partitioning-based strategy in CINEMA, the MAG-list can be maintained in a central machine and the maximization of influence for the subnetworks can be distributed into several machines and computed in parallel. Hence, as we shall see later, CINEMA can not only be realized on a single machine but also easily be adopted on a distributed platform by leveraging the *MapReduce* paradigm [12]. This further reduces the computation time of seed set.

In summary, the key contributions of this paper are as follows.

- We discuss the roles of influentials and conformers in the context of social influence analysis and show how to quantify influence and conformity of each individual in a social network. Specifically, we propose an iterative algorithm called CASINO (**C**onformity-**A**ware **S**ocial **IN**fluence c**O**mputation) that utilizes signs of social interactions (edges) to compute the influence and conformity indices of each node in an arbitrary social network.
- To the best of our knowledge, this is the first IM approach that leverages conformity of nodes to generate superior quality seeds. It is based on a novel cascade model called *conformity-aware cascade model* (C^2) which provides a formal framework to obtain the influence probabilities by taking into account the indices of nodes computed from CASINO. Moreover, we propose a *context-aware* C^2 model, called C^3 model, that takes into account the context-specific influence and conformity of nodes.
- For the first time, we systematically demonstrate the connection between cascade models deployed in IM problems and social psychology. Specifically, we prove that both the C^2 and C^3 models are consistent with the popular *social forces principle* [26] as they can degenerate to the latter. We also show how these models can be extended to align with the social forces principle by incorporating social forces exhibited by nodes that are more than a hop away from a given node (e.g., influence of friend-of-friend in *Twitter* or *Facebook*).
- Departing from several existing centralized, "non-partitioning-based" solutions to the IM problem, we propose a novel approach that addresses this problem by partitioning the underlying network into a set of non-overlapping subnetworks using an existing network partitioning technique and distributing influence spreads computation to relevant subnetworks. Specifically, we present a greedy algorithm called CINEMA that efficiently exploits the MAG-list build on top of the partitioned subnetworks to compute the seed set for influence maximization under our proposed model while maintaining superior quality of the influence spread. Importantly, CINEMA *produces superior quality seed set compared to existing greedy techniques without compromising on the computation cost.* Note that our solution ensures that CINEMA is not tightly coupled to any specific partitioning or conformity computation technique. This enhances generality as well as portability of CINEMA as it can be easily realized on top of a superior graph partitioning or conformity computation approach.
- We demonstrate that due to its inherent characteristics, the CINEMA algorithm can be gracefully adopted on a distributed platform. We realize this on a *MapReduce* framework. This further improves the running time by an order of magnitude with the increase in number of slave machines.
- By applying CINEMA and its distributed variant to real social networks comprising of millions of nodes, we show its effectiveness and significant improvement of performance over state-of-the-art methods.

The rest of the paper is organized as follows. In Sect. 2, we review related work. In Sect. 3, we investigate how to quantify influence and conformity of nodes in a social networks. In Sect. 4, we present the *conformity-aware cascade models* to study the influence propagation process with respect to conformity. We formally introduce the *partitioning-based influence maximization problem* based on this model in Sect. 5 and present an overview of the proposed CINEMA algorithm. In Sects. 6 and 7, we discuss in detail various key steps of CINEMA. A distributed implementation of CINEMA on the *MapReduce* framework is presented in Sect. 8. Section 9 presents our exhaustive experimental evaluation. The last section concludes the paper. A shorter version of this work appeared in [29,30]. The notations used in this paper are given in Appendix A of ESM.

2 Related work

2.1 Greedy IM approaches

Pedro and Matt [34] proposed a probabilistic method to predict the number of influenced nodes in a network by adopting Markov random field to study the propagation of influence. Kempe et al. [23] proved that solving such a problem is NP-hard. Hence, they proposed an approximate greedy algorithm based on the fact that if a greedy maximization algorithm of a *submodular* function f returns the result A_{greedy}, then the following holds $f(A_{greedy}) \geq (1-1/e) \max_{|A| \leq k} f(A)$. That is, it can give near optimal solution to the problem of maximization of a submodular function. Accordingly, Kempe et al. guaranteed that their greedy algorithm can achieve influence spread within $(1-1/e)$ of the optimal influence spread. However, the proposed algorithm takes $O(knmR)$ time to solve the IM problem where n and m are number of vertices and edges in the network, respectively, and R denotes the number of rounds of simulation. Note that this is computationally very expensive for real-world social networks.

Leskovec et al. [28] proposed an algorithm called CELF (Cost-Effective Lazy Forward) that is reported to be 700 times faster than the one proposed by Kempe et al. It is also based on the submodular property of the cascade influence function. They observed that in each round, in most cases the *marginal gain* of a node v, given by $\sigma(v|S) = \sigma(S \cup \{v\}) - \sigma(S)$, may not change significantly between consecutive rounds. So instead of recomputing the spread for each node at every round of seed selection, CELF performs a *lazy* evaluation. In the worst case, during each selection CELF needs to recompute the marginal gain for all the remaining nodes resulting in a worst-case time complexity of $O(kmRn)$.

Chen et al. [8] reduced the computation of marginal gain from $O(mn)$ to $O(m)$. Since in IC model each edge has the probability p to take effect in the cascade, they randomly remove each edge in the graph G with probability $1 - p$. In this way, G is separated into pieces and each piece is the scope of the node v's influence spread within it. Thus, computing the marginal gain of a node will only require a linear traversal of the scope. Similarly, when the network follows the WC model, each edge is removed with probability $1-1/v.degree$. The influence of each node can be computed by adding the gain in R iterations of random removal process. Based on this, the authors proposed the *MixGreedy* algorithm which follows the random removal process in computing the marginal gains and then utilizes the CELF approach for updates. The time complexities of the *MixGreedy* approach for the aforementioned two cascade models are $O(kRm)$ and $O(kTRm)$, respectively, where T is the number of iterations in gain computation. They demonstrated that its running time is smaller than CELF.

Wang et al. [42] proposed a community-based greedy solution to the IM problem. In order to reduce the running time, they first detect communities based on the IC model and then mine the top-k nodes across communities. A cost function is proposed to optimize the community assignment in mobile networks. Particularly, the community detection process takes $O(m + nR\ell m' + k\ell Rm')$ where ℓ denotes the decrease in the number of communities after the community combination process. Consequently, it is time consuming in huge networks. Furthermore, this effort does not consider conformity of nodes and topic-awareness in influence propagation. Also, there is a lack of systematic study on whether it can be realized in a distributed environment.

2.2 Heuristic-based IM approaches

The running times of the aforementioned greedy approaches are still large and may not be suitable for very large social networks. Hence, Chen et al. [8] used *degree discount* heuristic, where each neighbor of newly selected seed discounts its degree by one, to improve the running time (time complexity is $O(klogn + m)$). More recently, they proposed PMIA technique [7] over IC model, which selects a limited number of paths that satisfy a given threshold θ to compute the influence. The authors demonstrated that PMIA improves the influence spread generated by degree discount by 3.9–6.6 % over *Hep* dataset [8]. However, the running time of PMIA is an order of magnitude slower than the degree discount-based technique with time complexity of $O(nt_{i\theta}+kn_{o\theta}n_{i\theta}(n_{i\theta}+\log n))$ where $t_{i\theta}$, $n_{i\theta}$, $n_{o\theta}$ are constants decided by θ. Jung et al. [22] proposed IRIE which estimates the influence spread using a set of linear equations and demonstrated that the influence spread quality is similar to that of PMIA with more stable running time. The LDAG model [9] is similar to PMIA except that it is specifically designed for the *linear threshold* model. More recently, Goyal et al. [18] proposed a heuristic-based approach called SIMPATH in order to improve the seed quality of LDAG by consuming less memory. However, unlike greedy algorithms, the quality of influence spread of these models is not guaranteed to be within 63 % of the optimal.

Jiang et al. [20] proposed a simulated annealing-based approach for the IC model. Specifically, two heuristic methods are proposed to accelerate the convergence process of the algorithm. It initiates the seeds set by randomly selecting k nodes. In each iteration afterward, a node in the current seed set is replaced by another one which are not in the seeds, thus a new seed set is formed. If the new seed set can generate better influence spread than the old one under IC model, the seed set is updated to the new one. This process is iterated for T times until it converges. The time complexity of the algorithm is $O(Tk\overline{d})$ where $\overline{d}$ denotes the average degree of nodes. Experimental results have shown that the two heuristic methods have similar running time to the *degree discount* algorithm but better influence spread quality. However, the improvement in result quality is limited (i.e., 3–8 %).

Chen et al. [6] proposed a model called IC-N which introduces a *quality factor* to control the *negative opinion propagation probability*. In order to maximize the influence under IC-N model, a heuristic algorithm called MIA-N, which borrows the core idea of PMIA, is developed. It uses the notion of *maximum influence in-arborescence* to estimate the influence to an arbitrary node v from other nodes. Although this approach incorporates negative opinions in networks, it assumes that each node have the same influence and consequently exhibits the same quality factor. However, in real social networks individuals may exhibit different probabilities to express opposite opinions. In fact, the quality factor to control the negative opinion propagation probability can be viewed as a special case of conformity where an individual negatively follows another. CINEMA addresses this problem by computing a pair of *influence* and *conformity* indices for each individual.

Recently, Kim et al. [24] proposed an approximate IM algorithm called IPA under IC model that efficiently approximates influence by considering an *independent influence path* (influence paths between two nodes) as an influence evaluation unit. A parallelized version of IPA using OpenMP metaprogramming framework was also proposed that fully utilizes multi-core CPU resources. Empirical results revealed that it can solve the IM problem with competitive processing time and less memory usage.

CINEMA differs from the aforementioned approaches in the following ways. Firstly, our IM technique leverages on the conformity of nodes (extracted from real data) to compute influence probability for estimating influence spread. Secondly, we partition the network into a set of non-overlapping subnetworks and distribute the conformity-aware IM problem to relevant subnetworks to compute the seed set. Note that the time and space complexities of CINEMA reduce significantly as it runs on subnetworks which are often significantly smaller in sizes compared to the entire network. In contrast, as existing techniques (except for [42]) are designed to take the entire network as input for influence maximization, all the greedy approaches result in high computation cost due to the gigantic size of many online social networks. In contrast to [42], instead of designing an IC model-aware community detection method, we adopt existing network partition models which not only can be applied to *all* cascade models but also exhibit significantly smaller time complexity ($O(m)$). Last but not the least, as we shall see later, CINEMA can find significantly better quality of seeds as it exploits both influence and conformity of nodes. Given the fact that companies may invest months or years in designing new products, it is paramount to find seeds that give them opportunity to influence *relevant* population. Even though existing heuristic-based approaches are significantly faster than greedy strategies, we believe that companies are willing to wait few hours to find superior quality seed set as it may have significant impact on the marketing of products and its profits.

2.3 Action log-based approaches

Goyal et al. [16] proposed a supervised IM model by learning the influence probability from *action logs*. An *action log* is a set of triples (u, a, t) which says user u performed action a at time t. The basic idea is that if user v takes action a and later on v's friend u does the same, then the authors assume that a has propagated from v to u. They present a learning algorithm that not only predict the probability for action propagation but also the time when an action is expected to be performed. Specifically, they introduced the notion of *influenceablity*, which is defined as the ratio between the number of actions for which we have evidence that the user was influenced, over the total number of actions that have been performed by the user. Such a ratio is learned from action logs for each individual user and then utilized to predict the action propagation. The action logs of an arbitrary social network are also leveraged to learn the topic-level influence for each node [40].

More recently, they [17] proposed a *credit distribution* (CD) *model* that leverages the historical action logs to estimate influence spread. It assigns "credits" to the possible influencers of a node u whenever u performs an action. The sophisticated variant of this model distinguishes between different influenceability of different users by incorporating a *user influenceability function*. It is defined as the fraction of actions that u performs under the influence of at least one of its neighbors (e.g., v) and is learnt from the historical log data.

Our approach differs from the above methods in the following ways. Firstly, influenceability learning requires existence of a large amount of historical action logs to compute influence probability as well as user influenceability. Unfortunately, historical action logs may not be available to end users in many real-world social networks. In contrast, CINEMA does not require any historical action logs to compute conformity of nodes. Secondly, the probability of action propagation relies on the influenceability of the *object* user who is to be influenced and independent of the *subject* user who is influencing others. In contrast, in our model, the probability of action propagation depends on *both* the object and subject users. This leads to relatively superior seeds set in CINEMA (an example is given in Appendix B of ESM). Thirdly, in [16] influenceability is leveraged to predict the node activation time for a propagation instead of influence maximization. Lastly, no systematic study has been carried out to relate the notion of influenceability to well-known conformity-related concepts in social psychology to support the validity of the model.

Barbieri et al. [5] extended IC and LT cascade models to be *topic-aware*. They assume each item propagated through the network is a mixture of hidden topics. Then, an expectation maximization method is adopted to learn the topic distribution given the item propagation logs. However, they only applied the learned topic on traditional IC and LT models without considering the impact of influence and conformity on the influence propagation process.

2.4 Conformity-related research

The notion of *conformity* originated in social psychology. It is a type of social influence involving a change in belief or behavior in order to fit in with a group [2–4]. This change is in response to real (involving the physical presence of others) or imagined (involving the pressure of social norms/expectations) group pressure. In social psychology, there has been extensive study on the issue of social conformity [2–4,10,14,19,39]. We are inspired by these conformity studies and utilize it for influence spread computation in online IM problem.

Recently, Tang et al. [41] undertook conformity influence analysis in large social networks and proposed a model to model users' actions and conformity at three different levels, namely individual, peer, and group. A distributed learning algorithm is presented to efficiently learn the proposed model. Specifically, the conformity is defined and computed using users' action histories (e.g., the number of actions for which a user conforms to another), whereas in CINEMA we analyze the sentiments expressed by individuals to compute individual conformities. Hence, similar to [16,17], the former cannot be utilized to compute conformity of individuals for applications where action logs may not be available to end users. Furthermore, the authors did not provide any evidence on how the proposed conformity measures relate to well-known conformity-related concepts in social psychology to justify the robustness of the definitions. Most importantly, unlike CINEMA, this effort does not address the conformity-aware IM problem.

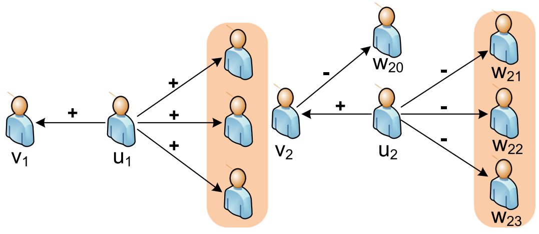
3 Influence and conformity computation

In this section, we formally introduce the notion of *influence* and *conformity* in the context of signed social networks and propose the CASINO algorithm to compute both indices. We begin by briefly introducing signed social networks, which lie at the foundation of our proposed strategy.

3.1 Signed social networks

Social interactions in online social networks can be either positive (indicating relations such as friendship) or negative (indicating relations such as distrust and opposition). For instance, in online discussion sites such as *Slashdot*, users can tag other users as "friends" (positive) and "foes" (negative). In blogosphere and *Twitter*, the reply relationship among users can be a positive or a negative one. In our following discussion, we treat such social interaction as signed directed graph.

In a *signed* social network $G(V, E)$, each edge has a positive or negative sign depending on whether it expresses a positive or negative attitude from the generator of the edge to the recipient [27]. Specifically, in this paper, a positive sign indicates that the recipient supports the opinion of the generator, whereas the negative sign represents otherwise. For example, Fig. 3b depicts a signed social network. The positive edge $E^+ = \{\overrightarrow{u_2 v_2}\}$ represents trust relationship while the negative ones ($E^- = \{\overrightarrow{w_{20} v_2}, \overrightarrow{u_2 w_{21}}, \overrightarrow{u_2 w_{22}}, \overrightarrow{u_2 w_{23}}\}$) represent distrust relationships. Note that the signs on the edges are not always available explicitly. In networks such as *Epinions* and *Slashdot*, the sign of each edge is explicitly provided.



(a) **(b)**

Fig. 3 Conformity and negative edge effect

However, in other networks such as blogosphere and *Twitter*, the sign of each edge is not explicitly available. In this case, we need to preprocess the network using text mining methods to discover signs associated with the links (detailed in Sect. 3.3). Consequently, a social network $G(V, E)$ containing both positive and negative edges can be represented using a pair of graphs $G^+(V, E^+)$ and $G^-(V, E^-)$ such that the following hold.

$$\forall \overrightarrow{uv} \in E, \begin{cases} \left(\overrightarrow{uv} \in E^+\right) \cap \left(\overrightarrow{uv} \in E^-\right) = 0, \\ \left(\overrightarrow{uv} \in E^+\right) \cup \left(\overrightarrow{uv} \in E^-\right) = 1 \end{cases}$$

In other words, $G^+(V, E^+)$ denotes the induced graph of positive edges E^+ (trust/agreement relationship) and $G^-(V, E^-)$ denotes that of negative edges E^- (distrust/disagreement relationship).

3.2 Influence and conformity indices

In our approach, each individual (vertex) in a signed network is associated with a pair of *influence index* and *conformity index* to describe the power of influence and conformity of the individual, respectively. Social psychologists have been using the term conformity to refer to the act of matching attitudes, beliefs, and behaviors to group norms since 1951 [4,14,26,39]. Researchers in social psychology have proposed principles modeling the ratio of individuals conforming to a group with respect to the group size, number of neighbors, etc. However, these works assume that individuals exhibit exactly the *same* conformity toward their neighbors, which may not necessarily be true in reality. Moreover, these approaches neither propose any quantitative method to evaluate the conformity of each individual nor take into account the negative conformity phenomenon, where an individual may explicitly break the group norms. In the following, we propose a novel algorithm to compute the conformity and influence indices for arbitrary social networks. Reconsider the signed network in Fig. 3b. Intuitively, the influence of v_2 should increase as *aggregated* conformity of those who trust v_2 (i.e., u_2) increases. On the other hand, the influence of v_2 should decrease if the aggregated conformity of those who distrust v_2 (i.e., w_{20}) increases. Thus, the *influence index* of an individual should capture this interplay of influence and conformity and penalize her whenever necessary.

Definition 1 (*Influence Index*) Let $G^+(V, E^+)$ and $G^-(V, E^-)$ be the induced graphs of the signed social network $G(V, E)$. The *influence index* of vertex $v \in V$, denoted as $\Phi(v)$, is defined as follows.

$$\Phi(v) = \sum_{\overrightarrow{uv} \in E^+} \Omega(u) - \sum_{\overrightarrow{uv} \in E^-} \Omega(u)$$

where $\Omega(u)$ represents the *conformity index* of vertex $u \in V$.

Similarly, the *conformity index* of u_2 in Fig. 3b depends on the influences of vertices which are trusted or distrusted by u_2. Intuitively, as the aggregated influence of those vertices which u_2 trust (e.g., v_2) increases, u_2 is more inclined to conform to others. On the other hand, when the aggregated influence of vertices which u_2 distrust (e.g., w_{21}, w_{22}, w_{23}) increases, u_2 is less inclined to conform to others. This intuition is inspired by research in social psychology which advocates that higher influence of an individual leads to higher conforming behaviors [11,36]. To elaborate further, suppose node u (resp. v) distrusts a group of neighbors, namely S_u (resp. S_v), consisting of k nodes. Assume that the aggregated influence of S_u is much larger than that of S_v. Since more influence can lead to a higher probability of action propagation [11], each node in S_u has a probability proportional to its influence to activate u. If we ignore the conformity of users, u (resp. v) will be activated with probability $1 - (1 - p_u)^k$ [resp. $1 - (1 - p_v)^k$] where p_u (resp. p_v) is the influence probability in S_u (resp. S_v). Obviously, u should have higher probability to be activated than v as $p_u \gg p_v$. If we consider this along with the fact that u and v exhibit different conformity and distrust S_u and S_v, respectively, then it is reasonable to assume that u is less inclined to conform to others compared to v.

Definition 2 (*Conformity Index*) Let $G^+(V, E^+)$ and $G^-(V, E^-)$ be the induced graphs of the signed social network $G(V, E)$. The *conformity index* of vertex $u \in V$, denoted as $\Omega(u)$, is defined as follows.

$$\Omega(u) = \sum_{\overrightarrow{uv} \in E^+} \Phi(v) - \sum_{\overrightarrow{uv} \in E^-} \Phi(v)$$

Thus, according to the above definition, the influence index of v_2 in Fig. 3b can be computed as $\Phi(v_2) = \Omega(u_2) - \Omega(w_{20})$. The conformity index of u_2 is computed as $\Omega(u_2) = \Phi(v_2) - \Phi(w_{21}) - \Phi(w_{22}) - \Phi(w_{23})$. Hence, the conformity of u_2 is positively affected by the influence of v_2 and negatively by the influences of w_{21}, w_{22}, and w_{23}. Thus, the above definition for conformity index is in accord with the intuition of conformity. Observe that the aforementioned definitions of influence and conformity are mutually dependent on each other. Consequently, a recursive computation framework is necessary to compute these two indices.

Algorithm 1: The CASINO algorithm.

```
Input: Social network $G(V, E)$
Output: the influence index
        $\mathbb{I}_\mathbb{T} = (\Phi_\mathbb{T}(u_1), \Phi_\mathbb{T}(u_2), \ldots, \Phi_\mathbb{T}(u_\ell))$ and conformity
        index $\mathbb{C}_\mathbb{T} = (\Omega_\mathbb{T}(u_1), \Omega_\mathbb{T}(u_2), \ldots, \Omega_\mathbb{T}(u_\ell))$ for
        $V = \{u_1, u_2, \ldots, u_\ell\}$ and for each topic $\mathbb{T}$
1 begin
2     $\mathcal{G} \leftarrow$ extractSubgraph($G$);
3     $\mathcal{G} = \{G\}$;
4     foreach $G_\mathbb{T} \in \mathcal{G}$ do
5         if $G_\mathbb{T}$ is not a signed network then
6             $(G^+_\mathbb{T}(V_\mathbb{T}, E^+_\mathbb{T}), G^-_\mathbb{T}(V_\mathbb{T}, E^-_\mathbb{T})) \leftarrow$ edgeLabel($G_\mathbb{T}$);
7         $(\mathbb{I}_\mathbb{T}, \mathbb{C}_\mathbb{T}) \leftarrow$
          indicesCompute($G^+_\mathbb{T}(V_\mathbb{T}, E^+_\mathbb{T}), G^-_\mathbb{T}(V_\mathbb{T}, E^-_\mathbb{T})$);
```

3.3 The algorithm CASINO

We begin by briefly describing the notion of *context-aware* and *context-free* signed social networks to represent real-world online networks.

Online social networks can be classified into *context-aware* and *context-free* networks. The former represent networks where the edges are associated with topics (context) as social interactions may often involve conversations on specific topics. For example, each conversation in *Twitter* is based on a specific topic. Figure 1 depicts interactions between three users on the topic `iPad 2`. The leftmost social network in Fig. 4 is another example of context-aware social network where an edge labeled as $\mathbb{T}_1, \mathbb{T}_2$ indicates that the pair of individuals communicate with each other on topics $\mathbb{T}_1$ and $\mathbb{T}_2$. On the other hand, interactions in context-free networks do not involve specific topics. For example, in *Epinions* and *Slashdot* individuals trust (distrust) each other regardless of any specific topic.

The CASINO (**C**onformity-**A**ware **S**ocial **IN**fluence c**O**m putation) algorithm is outlined in Algorithm 1 and consists of three phases, namely the *topic-based subgraph extraction* phase (Line 3), the *edge labeling* phase (Line 6), and the *indices computation* phase (Line 7). Figure 4 depicts an overview of the algorithm. Given a social network $G(V, E)$, if it is a context-aware network then the *topic-based subgraph extraction* phase extracts a set of subgraphs of G (denoted by $\mathcal{G}$) where each subgraph $G_\mathbb{T}(V_\mathbb{T}, E_\mathbb{T}) \in \mathcal{G}$ contains all the vertices and edges in G associated with a specific topic $\mathbb{T}$. Each subgraph $G_\mathbb{T}$ represents positive or negative attitudes of individuals toward opinions of others in G with respect to the topic $\mathbb{T}$. For instance, in Fig. 4, this phase generates three topic-based subgraphs, namely, $G_{\mathbb{T}_1}$, $G_{\mathbb{T}_2}$, and $G_{\mathbb{T}_3}$, for topics $\mathbb{T}_1$, $\mathbb{T}_2$, and $\mathbb{T}_3$, respectively. Recall that edges of a social network may not be explicitly labeled with positive or negative signs. This is especially true for context-aware networks (e.g., *Twitter*). In contrast, links in many context-free

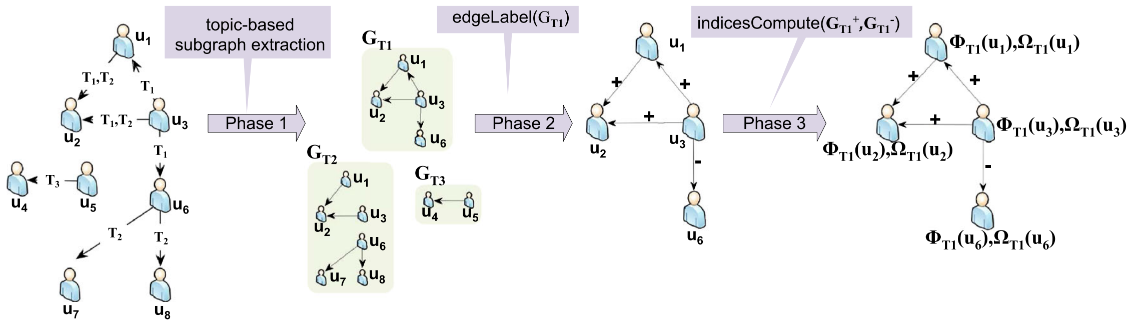


Fig. 4 Overview of CASINO

networks (e.g., *Slashdot* and *Epinions*) are explicitly labeled with signs. Hence, it is important to label the edges in each topic-based subgraph $G_{\mathbb{T}}$. The objective of the *edge labeling* phase is to assign a sign to each edge by analyzing the sentiment expressed by the generator and recipient of the edge. Figure 4 depicts the labeling of $G_{\mathbb{T}_1}$. Finally, given a set of signed topic-based subgraphs $\mathcal{G}$, the goal of the *indices computation* phase is to iteratively compute the influence and conformity indices of each individual in each $G_{\mathbb{T}} \in \mathcal{G}$. Observe that a vertex v in G may have multiple pairs of indices if v is involved in more than one topic-based subgraph. Since the first phase is straightforward, we describe the remaining two phases.

Algorithm 2: The *edgeLabel* procedure.

```
Input: Topic-based subgraph $G_{\mathbb{T}}(V_{\mathbb{T}}, E_{\mathbb{T}})$ induced by topic $\mathbb{T}$,
Output: $G^+_{\mathbb{T}}(V_{\mathbb{T}}, E^+_{\mathbb{T}})$ and $G^-_{\mathbb{T}}(V_{\mathbb{T}}, E^-_{\mathbb{T}})$ such that:
        $E^+_{\mathbb{T}} \cup E^-_{\mathbb{T}} = E_{\mathbb{T}}$ and $E^+_{\mathbb{T}} \cap E^-_{\mathbb{T}} = \emptyset$
1 begin
2     $E^+_{\mathbb{T}} = E^-_{\mathbb{T}} = \emptyset$;
3     foreach $\overrightarrow{u\mathbb{T}v} \in E_{\mathbb{T}}$ do
4         $u.sentiment \leftarrow LingPipe.$sentExtr$(u)$;
5         $v.sentiment \leftarrow LingPipe.$sentExtr$(v)$;
6         if $|u.sentiment - v.sentiment| < \epsilon$ then
7             $E^+_{\mathbb{T}} = E^+_{\mathbb{T}} \cup \{\overrightarrow{u\mathbb{T}v}\}$
8         else
9             $E^-_{\mathbb{T}} = E^-_{\mathbb{T}} \cup \{\overrightarrow{u\mathbb{T}v}\}$
```

The edge labeling phase In this paper, we adopt the method described in Algorithm 2. We denote each edge $\overrightarrow{uv}$ associated with topic $\mathbb{T}$ as $\overrightarrow{u\mathbb{T}v}$. This enables us to differentiate between an edge which shares the same generator and recipient for more than one topic. For each edge $\overrightarrow{uAv}$ in a topic-based subgraph $G_{\mathbb{T}}$, we identify 5-leveled sentiment (i.e., like, somewhat like, neutral, somewhat dislike, dislike) expressed at both ends using *LingPipe* [1], a popular sentiment mining package adopted in several recent studies [21,32,43] (Lines 4–5). Note that *LingPipe* has been tested to provide very promising results (i.e., accuracy over 85 % in most cases [21,32]) on sentiment extraction. If the sentiments at both ends are similar (*sentiment similarity threshold* is less than ϵ), we denote the edge as positive (Lines 6–7). Otherwise, we denote it as negative (Lines 8–9).

Indices computation phase Given a topic $\mathbb{T}$ and topic-based subgraph $G_{\mathbb{T}}$, the preceding phase generates $G^+_{\mathbb{T}}$ and $G^-_{\mathbb{T}}$. Without loss of generality, assume that there are $|\mathcal{G}|$ different topics. Then, we are able to compute an individual's influence and conformity indices for each topic [(i.e., $\Phi_{\mathbb{T}}(u)$ and $\Omega_{\mathbb{T}}(u)$]. We now elaborate on the algorithm for computing these indices.

Algorithm 3 outlines the strategy for computing a pair of influence and conformity indices $(\Phi(u),\ \Omega(u))$ for each vertex u. It first initializes the influence index and conformity index of all vertices to be 1 (Lines 1–4).[1] Subsequently, in each iteration, it computes them for each vertex by using the values of the indices in previous iteration (Lines 6–8) and normalizing these values using the square root of the summation of all vertices' index values (Lines 9–13). The algorithm terminates when all indices converge. We shall now prove that the proposed algorithm is guaranteed to converge after a fixed number of iterations n. In other words, the difference between an arbitrary node's indices between n and $n+1$ rounds of iteration is insignificant, and hence, we do not need to consider additional iterations.

Theorem 1 *The indicesCompute procedure described in Algorithm 3 converges.*

Proof The proof is given in Appendix C of ESM. □

During each iteration, Algorithm 3 traverses all the edges attached to each node. Hence, the complexity for running an iteration take $O(m'n')$ time, where m' (resp. n') is the maximum number of edges (resp. nodes) in a subgraph. Without loss of generality, assume Algorithm 3 converges after k' iterations. Thus, time complexity of Algorithm 3 is $O(k'm'n')$.

Observe that the aforementioned technique can easily be extended to compute the aggregated indices of an individual by taking into account the entire social network G over

[1] We have experimented with different initial values. All strategies converge to the same results with different number of iterations (see Appendix D of ESM for details).

Algorithm 3: The *indicesCompute* procedure.

```
Input: $G(V, E) = G^+(V, E^+) \cup G^-(V, E^-)$
Output: the influence index $\mathbb{I} = (\Phi(u_1), \Phi(u_2), \ldots, \Phi(u_\ell))$ and
        conformity index $\mathbb{C} = (\Omega(u_1), \Omega(u_2), \ldots, \Omega(u_\ell))$ for
        $V = \{u_1, u_2, \ldots, u_\ell\}$
begin
    $k = 1$ /*initialize iteration counter*/ ;
    foreach $u \in V$ do
        $\Phi^k(u) = \Omega^k(u) = 1$
    while $\mathbb{I}$ or $\mathbb{C}$ not converged do
        foreach $u \in V$ do
            $\Phi_0^{k+1}(u) = \sum_{\overrightarrow{vu} \in E^+} \Omega^k(v) - \sum_{\overrightarrow{vu} \in E^-} \Omega(v)$;
            $\Omega_0^{k+1}(u) = \sum_{\overrightarrow{uv} \in E^+} \Phi^k(v) - \sum_{\overrightarrow{uv} \in E^-} \Phi(v)$;
        foreach $u \in V$ do
            $\Phi^{k+1}(u) = \frac{\Phi_0^{k+1}(u)}{\sqrt{\sum_{v \in V} \Phi_0^{k+1}(v)^2}}$;
            $\Omega^{k+1}(u) = \frac{\Omega_0^{k+1}(u)}{\sqrt{\sum_{v \in V} \Omega_0^{k+1}(v)^2}}$;
        $\mathbb{I}^{k+1} = (\Phi^{k+1}(u_1), \Phi^{k+1}(u_2), \ldots, \Phi^{k+1}(u_l))$;
        $\mathbb{C}^{k+1} = (\Omega^{k+1}(u_1), \Omega^{k+1}(u_2), \ldots, \Omega^{k+1}(u_l))$;
        $k = k + 1$;
```

all topics $\mathbb{T} = 1, \ldots, |\mathcal{G}|$. In this case, E^+ and E^- in Definitions 1 and 2 are replaced by $\bigcup_{\mathbb{T}=1}^{|\mathcal{G}|} E_{\mathbb{T}}^+$ and $\bigcup_{\mathbb{T}=1}^{|\mathcal{G}|} E_{\mathbb{T}}^-$, respectively.

4 Conformity-aware cascade model

In the preceding section, we have described how to quantify conformity behavior of individuals in a social network. In this section, we formally introduce a novel cascade model that takes into account conformity of nodes for influence propagation. We begin by briefly describing the classical influence maximization (IM) problem that has been considered in the literature.

4.1 Classical influence maximization problem

A social network is modeled as directed graph $G = (V, E)$, where nodes in V modeling the individuals in the network and edges in E modeling the relationship between them. The influence maximization (IM) problem is defined as follows [23].

Definition 3 (*Influence Maximization Problem*) Given a social network $G(V, E)$, a specific cascade model C and a budget number k, the **influence maximization (IM) problem** is to find a set of nodes S in G, which we call as seed set, where $|S| = k$ such that according to C, the expected number of nodes that are influenced by S [denoted by $\sigma(S)$] is the largest. It can be expressed as follows:

$$S = \underset{S' \subseteq V, |S'| = k}{\arg\max} \ \sigma(S')$$

Note that cascade model refers to the model that defines how a piece of information propagates from an individual to another in the network. Majority of the literature on influence maximization have focused on the *independent cascade* (IC), *weighted cascade* (WC), and *linear threshold* (LT) models [23] (see Appendix E of ESM for details). The optimum solution to the IM problem is NP-hard for the aforementioned cascade models [23]. However, as remarked earlier, greedy approximation algorithms exist for the optimal solution to be approximated to within a factor of $(1 - 1/e)$ as long as the influence function $\sigma(\cdot)$ is *submodular*. Let S be a finite set. Then, a function $f : 2^S \rightarrow R$ is *submodular* if $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for $\forall A \subseteq B \subseteq S$ and $v \in S$. In another word, the *marginal gain* from adding an element to a set A is at least as much as the *marginal gain* from adding the same element to a superset of A. In the case of IM problem, $\sigma(\cdot)$ is submodular, takes only nonnegative values, and is monotone in the sense that adding an element to a set cannot cause f to decrease. The *marginal gain* of a node v given the seed set S is defined as following [23].

Definition 4 (*Marginal Gain*) Given a cascade model C, a node v, and the current seed set S, the **marginal gain** of v with respect to S, denoted by $\sigma(v|S)$, is defined as $\sigma(v|S) = \sigma(S \cup \{v\}) - \sigma(S)$. That is, $\sigma(v|S)$ denotes the increase in the expected number of nodes that are influenced due to the addition of v in S.

Greedy solution toward IM problem works by iteratively selecting the node which shows the most marginal gain for current S. Thus, each time after adding a node into S, the greedy algorithm has to update each node's marginal gain for current S and select the one with the maximum marginal gain.

4.2 Conformity-aware cascade model

Recall that existing IM techniques do not leverage conformity of nodes for computing influence probabilities. We also showed that the presence of an edge between a pair of node u and v is highly affected by the influence of u and the conformity of v. Thus, the probability of influence propagation from u to v is affected by not only influence of u but also conformity of v. Inspired by this finding, we define the *conformity-aware cascade* (C^2) model as follows.

Definition 5 (C^2 *Model*) Let A_i be the set of nodes that are influenced in the i-th round and $A_0 = S$. Let M be the adjacency matrix of a social network $G(V, E)$. For any $(u, v) \in E$ (i.e., $M[u, v] \neq 0$) such that u is already in A_i and v is not yet influenced, v is influenced by u in the next $(i + 1)$-th round with a probability that is proportional to the product of u's influence [denoted by $\Phi(u)$] and v's conformity [denoted by $\Omega(v)$]. Thus, the probability $v \in A_{i+1}$ can be computed as:

$$1 - \prod_{u \in A_i, M[u,v] \neq 0} (1 - \Phi(u)\Omega(v))$$

This process is repeated until A_{i+1} is empty.

A keen reader may observe that in contrary to existing cascade models, the "activation probability" in Definition 5 is defined as multiplication of $\Omega(\cdot)$ and $\Phi(\cdot)$. Consequently, it may seem that the number of influenced users should be smaller in this model. However, as we shall see in Sect. 9.2, the final activated number of users is surprisingly not as small as expected. This is primarily due to the characteristics of the seeds selected by IM algorithms (detailed in Sect. 9.2).

The C^2 model is suitable for networks where the influence and conformity indices are independent of any specific topic of discussion (e.g., *Epinions*, *Hep*). Hence, it is applicable for context-free networks as interactions in such networks do not involve specific topics. However, in a context-aware network, the influence and conformity of nodes are topic-dependent. Thus, we need to extend C^2 model so that it can be applied to such networks where the influence and conformity indices are correlated with specific topics. We refer to this model as *conformity and context-aware cascade* (C^3) model and is formally defined as follows.

Definition 6 (C^3 *Model*) Let A_i be the set of nodes that are influenced by topic $\mathbb{T}$ in the i-th round and $A_0 = S$. Let $M_{\mathbb{T}}$ be the adjacency matrix of a social network $G_{\mathbb{T}}(V, E_{\mathbb{T}})$. For any $(u, v) \in E_{\mathbb{T}}$ (i.e., $M_{\mathbb{T}}[u, v] \neq 0$) such that u is already in A_i and v is not yet influenced, v is influenced by u in the next $(i + 1)$-th round with a probability that is proportional to the product of u's influence [denoted by $\Phi_{\mathbb{T}}(u)$] and v's conformity [denoted by $\Omega_{\mathbb{T}}(v)$]. Thus, the probability $v \in A_{i+1}$ can be computed as: $1 - \prod_{u \in A_i, M_{\mathbb{T}}[u,v] \neq 0} (1 - \Phi_{\mathbb{T}}(u)\Omega_{\mathbb{T}}(v))$. This process is repeated until A_{i+1} is empty.

Reconsider the example in Sect. 1.1. Recall that v_5 (resp. v_3) was selected as a seed under the IC (resp. WC) model. However, this may not be true if we take into account the conformity of nodes or topics-related information in the network. The influence and conformity indices of each node is listed in Table 2 (fourth and fifth columns). Based on Definition 6, $Prob[\overrightarrow{v_3 v_4} \notin X]$ can be computed as $1 - (\Phi_1(v_3)\Omega_1(v_4)) = 0.51$. Thus, $\sigma(v_3) = 0.51 \times 1 + 0.49 \times 2 = 1.49$. The expected influences of the remaining nodes with respect to the topic "`iPad`" under C^3 model are listed in Table 1 (forth row). Thus, we should select v_1 (instead of v_5 or v_3) as the seed when we consider conformity of nodes in a context-aware network. We shall validate our hypothesis empirically in Sect. 9.

Theorem 2 *Given a social network graph $G(V, E)$, the influence function $\sigma(\cdot)$ under* C^2 *(resp.* C^3*) model is submodular.*

Proof The proof is given in Appendix F of ESM. □

4.3 Implication and interpretation of C^2 and C^3 models in social psychology

Existing cascade models (e.g., IC, LT) are defined specifically for online social networks and are tuned to support algorithms designed for measuring influence. Surprisingly, to the best of our knowledge, there is no systematic study to connect such models to real-world principles related to social influence which are endorsed by the social psychology community. Hence, a key issue related to our proposed conformity-aware models is that whether they are aligned with well-known conformity-related principles in social psychology? In this section, we provide answer to this question affirmatively.

In social psychology, there has been a host of study investigating *social impact* [26] between individuals within a social network. A well-known principle of social impact states that when some number of social sources are acting on a target individual, the amount of *social forces* experienced by the target should be a multiplicative function of the *strength* S, the *immediacy* I, and the number of sources N present [26]. That is, $F = f(S \times I \times N)$. Here, the strength S represents the power, importance, or intensity of a source to the target and the immediacy I represents the closeness in space or time. In the following, we prove that both our proposed C^2 and C^3 models are consistent with this social forces principle by degenerating these models to the aforementioned principle, $F = f(SIN)$.

According to Definition 5, v is influenced with probability $1 - \prod_{u \in A_i, M[u,v] \neq 0} (1 - \Phi(u)\Omega(v))$. In fact, the probability of v to be influenced can be viewed as a variation of social forces v experienced, thus,

$$1 - F = \prod_{u \in A_i, M[u,v] \neq 0} (1 - \Phi(u)\Omega(v)).$$

Hence,

$$\ln(1 - F) = \sum_{u \in A_i, M[u,v] \neq 0} \ln(1 - \Phi(u)\Omega(v)).$$

Let α_j, β denote $\Phi(u)$, $\Omega(v)$ respectively, then the above equation is $\ln(1 - F) = \sum_{j=1}^{N} \ln(1 - \alpha_j \beta)$, where $N = |\{u | u \in A_i, M[u, v] \neq 0\}|$. Thus,

$$F = 1 - \exp\left(\sum_{j=1}^{N} \ln(1 - \alpha_j \beta)\right).$$

If we assume all individuals exhibit similar influence index α then the above equation degenerates to the following form

$$F = 1 - \exp(N \ln(1-\alpha\beta)).$$

Observe that the above equation is consistent with social forces principle $F = f(SIN)$ as $\ln(1-\alpha\beta)$ in fact evaluates the importance and intensity of a source to the target, namely S. Hence, our models are consistent with the social forces principle in social psychology.

4.4 Immediacy-aware cascade models

Observe the differences between the social forces principle F and our proposed models. Firstly, both C^2 and C^3 models assume that different individuals exhibit *different* degree of influence on the target node, which is more realistic in real-world social networks. Secondly, they only consider source nodes which are immediate neighbors to the target node limiting the immediacy I to 1. That is, we do not consider nodes that are more than a hop away from the target node ($I \geq 1$). At first glance, it may seem that such assumption is justified in online social networks as two nodes that are more than one hop away from each other may not have influence between them. Interestingly, this may not be true always as it is indeed possible to be influenced by individuals more than a hop away. For example, in social networking sites such as *Facebook* and *Twitter*, it is possible for one to view a friend-of-friend's comments or posts (depending on the privacy settings) and be influenced although there is no direct friendship (edge) between them. Indeed, similar observation is reported by recent studies such as [33] where the authors define *n-degree influence* as $p_1 - p_2$, where p_1 is the probability that u is activated given that its n-hop-away friend v is activated and p_2 is the average probability of activation over all nodes. They showed using Twitter data that such "indirect" influence is not trivial. However, these models are neither grounded on social forces principle nor are conformity-aware. Hence, we now extend C^2 and C^3 models by incorporating effects of nodes with $I \geq 1$.

Definition 7 [IC^2 *Model*) Let A_i be the set of nodes that are influenced in the i-th round and $A_0 = S$; let M be the adjacency matrix of a context-free social network $G(V, E)$. For any $(u, v) \in E$ (i.e., $M[u, v] \neq 0$) such that u is already in A_i and v is not yet influenced, v is influenced by u in the next $(i+1)$-th round with a probability that is proportional to $\Phi(u)\Omega(v)$. Moreover, for any node $u' \in A_j, (j < i)$ and $M^{i-j}[u', u] \neq 0$ (i.e., u is reachable from u' which is $(i-j)$-hops away), v is influenced by u' with a probability $e^{j-i}\Phi(u')\Omega(v)$. Hence, v is influenced by u and all u' with a probability $\sum_{M^{i-j}[u',u]\neq 0} e^{j-i}\Phi(u')\Omega(v)$. Especially, when $j = i$, the matrix M^{i-j} is an identity matrix and $M^{i-j}[u', u] \neq 0$ if and only if $u' = u$. Thus, the probability $v \in A_{i+1}$ can be computed as:

$$1 - \prod_{u\in A_i, M[u,v]\neq 0}\left(1 - \sum_{M^{i-j}[u',u]\neq 0,\, j\in[0,i]} e^{j-i}\Phi(u')\Omega(v)\right).$$

This process is repeated until A_{i+1} is empty.

In fact, the above model is a more generalized form of the C^2 model which takes into account social forces exhibited by nodes that are more than 1-hop away. Hence, it is identical to the social forces principle $F = f(SIN)$ as e^{j-i} can be interpreted as the immediacy between source and target nodes. In particular, the IC^2 model degenerates to the C^2 model when the distance between u, v [i.e., $(i+1-j)$] is 1 (i.e., $i = j$).

Similarly, it is intuitive to generalize the C^3 model to IC^3 model by taking into account the immediacy in context-aware subgraphs as follows.

Definition 8 (IC^3 *Model*) Let A_i be the set of nodes that are influenced by topic $\mathbb{T}$ in the i-th round and $A_0 = S$; let $M_{\mathbb{T}}$ be the adjacency matrix of a context-aware social network $G_{\mathbb{T}}(V, E_{\mathbb{T}})$. For any $(u, v) \in E$ (i.e., $M_{\mathbb{T}}[u, v] \neq 0$) such that u is already in A_i and v is not yet influenced, v is influenced by u in the next $(i+1)$-th round with a probability that is proportional to $\Phi_{\mathbb{T}}(u)\Omega_{\mathbb{T}}(v)$. Moreover, for any node $u' \in A_j, (j < i)$ and $M_{\mathbb{T}}^{i-j}[u', u] \neq 0$ (i.e., u is reachable from u' which is $(i-j)$-hops away), v is influenced by u' with a probability $e^{j-i}\Phi_{\mathbb{T}}(u')\Omega_{\mathbb{T}}(v)$. Hence, v is influenced by u and all u' with a probability $\sum_{M_{\mathbb{T}}^{i-j}[u',u]\neq 0} e^{j-i}\Phi_{\mathbb{T}}(u')\Omega_{\mathbb{T}}(v)$. Thus, the probability $v \in A_{i+1}$ can be computed as:

$$1 - \prod_{u\in A_i, M_{\mathbb{T}}[u,v]\neq 0}\left(1 - \sum_{M_{\mathbb{T}}^{i-j}[u',u]\neq 0,\, j\in[0,i]} e^{j-i}\Phi_{\mathbb{T}}(u')\Omega_{\mathbb{T}}(v)\right)$$

This process is repeated until A_{i+1} is empty.

Observe that the IC^2 (resp., IC^3) model theoretically demonstrates that it is possible to generalize the C^2 (resp., C^3) model, if necessary, to align it to the aforementioned social forces principle. Interestingly, in Sect. 9, we shall empirically demonstrate that in online social networks both these types of models tend to generate similar seed sets. That is, the impact of nodes more than one hop away is not significant in representative networks as far as seed set quality is concerned. Thus C^2 (resp., C^3) model is a good approximation of the IC^2 (resp., IC^3) model in IM task by removing unnecessary computation on distant neighbors. It is worth noting that given the dynamic nature of social networks, we may leverage IC^2 (resp., IC^3) model if the influence of friend-of-friend becomes significant (especially in the context of certain topics).

In summary, our IC^2 (resp., IC^3) model is not only well-aligned with social forces principle in social psychology but collectively with C^2 (resp., C^3) model it gives us the flexibility to use appropriate model for the IM problem depending on the influence characteristics of the underlying dynamic network.

5 Overview of CINEMA

We now have all the machinery in place to facilitate conformity-aware influence maximization in social networks. In this section, we give an overview of key steps of the Algorithm CINEMA, designed toward this goal. For ease of exposition, we assume that the social network is context-free. Hence, the topic-based subgraph extraction phase of CASINO is disabled. We begin by formally define the *partitioning-based influence maximization* problem.

5.1 Partitioning-based IM problem

Existing greedy approximation algorithms consume significant time on updating the marginal gains of the top nodes in the list and their rearrangements [8,23,28]. Hence, avoidance of unnecessary updates of marginal gains along with reduction of the size of the node list can reduce the computation cost significantly. We achieve this by taking a partitioning-based approach where the whole social network is partitioned into a set of non-overlapping subnetworks. By doing so, we ensure that changes to the marginal gain of a node in a subnetwork G_i do not affect nodes in another subnetwork G_j. Hence, the update of the marginal gains of nodes in G_i is restricted within it instead of the entire network. In fact, as we shall see later, the computation time of the update operation is reduced by a factor of m/m_i where m_i represent the number of edges in G_i.

Definition 9 (*Partitioning-based Influence Maximization Problem*) Given a budget k and a social network $G(V, E)$, let $\Gamma = \textbf{Partition}(G)$ be the partitions of G containing a set of subnetworks where $V = V_1 \cup V_2 \cup \cdots \cup V_{|\Gamma|}$, $V_i \cap V_j = \emptyset \forall i \neq j$, $0 \leq i, j < |\Gamma|$, and $(u, v) \notin E$ for $\forall u \in V_i, v \in V_j$. Let each G_i exhibits a specific cascade model C_i. Then the **partitioning-based influence maximization problem** finds a set of seeds S in Γ where $|S| = \sum_{i=1}^{|\Gamma|} |S_i| = k$ such that the expected number of nodes that are influenced by S is the largest in G. That is,

$$S = \underset{\sum |S_i| = k}{\arg\max} \sum_{S_i \subseteq V_i} \sigma(S_i)$$

Observe that in the aforementioned definition we theoretically generalize the problem by adopting different influence models in different subnetworks. Clearly, it can also handle the case where different subnetworks have the same cascade model (e.g., C^2 model) to reflect many real-world applications.

Algorithm 4: The CINEMA algorithm.

```
Input: Graph $G(V, E)$, budget $k$, and the cascade influence
       function $\sigma(\cdot)$
Output: Seed set $S$ of nodes, $|S| = k$
1 begin
2     $\Gamma \leftarrow$ NetworkPartition($G$);
3     foreach $G_i \in \Gamma$ do
4         $(G_i, (\Phi_i(\cdot), \Omega_i(\cdot))) \leftarrow$ ComputeConformity($G_i$) /*
          Alg. 1 */;
5     $(\mathcal{M}, \Upsilon) \leftarrow$ MAGConstruction($\Gamma, \sigma(\cdot), \Phi_i(\cdot), \Omega_i(\cdot)$);
6     $S \leftarrow$ SeedsSelection($G, k, \sigma(\cdot), \mathcal{M}, \Upsilon, \Phi_i(\cdot), \Omega_i(\cdot)$);
```

Theorem 3 *Given the social network graph $G(V, E)$ and $\Gamma = \textbf{Partition}(G)$, if the influence function $\sigma_i(\cdot)$ for each of the cascade model C_i of $G_i \in \Gamma$ is submodular, then $\sigma(S)$ in Definition 9 is also submodular.*

Proof The proof is given in Appendix G of ESM. □

Observe that Theorem 3 states that if the influence functions within each partition are submodular, then we have the usual $(1-1/e)$ guarantee for the solution quality for the partitioned network. Obviously, due to edge cuts during partitioning, it does not indicate that the partitioning-based solution will have the $(1 - 1/e)$ guarantee for the original optimization problem defined on the *whole* network. In spite of this, as we shall see later, our empirical results on variety of real social networks demonstrate that CINEMA consistently produces superior quality spreads compared to the conventional greedy approaches having $(1 - 1/e)$ guarantee.

Finally, we propose a theoretical analysis of how an arbitrary node's influence spread quality is affected in a partitioning-based IM method compared to non-partitioning approaches. The main factor that affects the influence spread quality in partitioning-based method is that some edges may be cut such that a group of nodes' expected influence may thus be discounted.

Theorem 4 *Let $\chi(V)$ be the maximal loss of expected influence of a node v in $G(V, E)$ due to the employment of a partitioning-based* IM *technique [*i.e., $\chi(V) = \max_{v \in V} (\sigma^0(v) - \sigma(v))$*]. Then*

$$\max_{m_i \in \Delta} p_{m_i} \sigma(v_i^e) \leq \chi(V) \leq \sum_{m_i \in \Delta} p_{m_i} \sigma(v_i^e)$$

where $\sigma^0(v)$ is the expected influence of v if partitioning is not employed, Δ is the set of cut edges and v_i^e is the end node of a cut edge $m_i \in \Delta$.

Proof The proof is given in Appendix H of ESM. □

5.2 Algorithm CINEMA

The CINEMA algorithm is outlined in Algorithm 4 and consists of four phases, namely the *network partitioning* phase (Line

2), the *conformity computation phase* (Lines 3–4), the MAG-*list construction* phase (Line 5), and the *seeds selection* phase (Line 6).

Phase 1: The network partitioning phase For any cascade model, influence always flows along edges in the social network graph. Hence, if there is no path between two nodes then it is not possible for influence to flow between these nodes. In this phase, we first partition the social network graph to a set of non-overlapping connected components (also referred to as subnetworks). As each component is unconnected to another component, the influence computation in a subnetwork is not affected by other subnetworks or components.

Note there are several existing techniques to generate disjoint dense connected components from a graph efficiently [38]. We take the BFS (Breadth First Search)-based strategy to traverse the graph and extract the connected components. The running time of this process is $O(m+n)$. Note that some real-world networks (e.g.,*LiveJournal*) are highly clustered and cannot be easily separated into a set of non-overlapping subnetworks using the BFS technique. Particularly, the BFS-based method may generate components having $m' \approx m$ for these networks. In this case, we partition the network into non-overlapping components using a ℓ-way partitioning algorithm provided by CLUTO[2] [38]. In Sect. 9, we shall justify choosing this graph partitioning algorithm over several existing ones. Given the number of partitions ℓ as input, it can provide good quality partitions in $O(m)$ time. Note that such partitioning process may inevitably remove some edges in the network. However, as graph partitioning algorithms often minimize the size of edge cuts, the removal of edges does not have significant adverse effect on the estimation of influences of nodes in comparison to existing greedy approaches. In fact, our experimental results in Sect. 9 demonstrate that for these networks CINEMA can still preserve high-quality seed set.

In summary, we undertake the following strategy for partitioning the social network graph. If the network can be easily clustered into non-overlapping components by BFS-based method such that $m' \ll m$, then we create the final subnetworks based on this strategy. However, if the BFS-based method fails to generate disjoint components or there exists components after partitioning such that $m' \approx m$, then we adopt the ℓ-way partitioning technique to generate the set of non-overlapping subnetworks.

Phase 2: The conformity computation phase In this phase, we use the CASINO algorithm (Sect. 2.4) to compute the influence and conformity indices of the nodes in each subnetwork generated from the preceding phase. Note that these indices will be used to compute the influence probabilities based on our conformity-aware cascade models. It is worth mentioning that CINEMA is not tightly coupled to any specific conformity computation technique and as a result its benefits can be realized on any superior conformity computation approach.

Phase 3: The MAG-*list construction phase* In contrast to the strategy of lazily updating the marginal gains of nodes existing in a single set, in CINEMA, the update of marginal gains needs to be carried out within each node set representing each subnetwork *independently*. Given that there may be a large number of subnetworks, how can we efficiently perform the update operations? Inspired by [8,28], in this phase, we construct two data structures, namely MAG-*list* and a set of COG-*sublist*s over the subnetworks that enable us to efficiently determine which subnetwork the next seed should be selected from and how to effectively perform updates of marginal gains across subnetworks. Informally, a MAG-*list* contains nodes with maximum marginal gain in the subnetworks. Each COG-*sublist i*s associated with a subnetwork or component and stores the marginal gains of all nodes in the subnetwork. We shall elaborate on this phase in Sect. 6.

Phase 4: The seeds selection phase Lastly, this phase exploits the MAG-list to compute the seed set from the set of subnetworks (Sect. 7). It iteratively selects the node having maximum marginal gain from the MAG-list and, if necessary, efficiently updates and reorders nodes in relevant COG-sublists dynamically.

6 MAG-list construction

In this section, we present the MAG-list (**MA**rginal **G**ain List) data structure which we shall be exploiting for the influence maximization problem. We begin by introducing the notion of ***component gain sublist*** (COG-sublist) which we shall be using to define MAG-list. Given a subnetwork $G_i(V_i, E_i)$ where $G_i \in \mathit{\Gamma}$, the *component gain sublist* of G_i, denoted by β^i, contains the list of nodes V_i. Each node $v \in \beta^i$ and $v \in V_i$ is a 3-tuple $(ID, gain, valid)$ where ID is the unique node identifier of v in G, $gain$ is the marginal gain with respect to S_i, and $valid$ is a boolean variable indicating whether the marginal gain of v is up-to-date. The list is sorted in descending order based on the marginal gains of the nodes. Hence, the node with maximum marginal gain is the top element in the sublist, denoted by $top(\beta^i)$. The *size* of COG-sublist is denoted by $|\beta^i| = |V_i|$. Note that since a social network graph is partitioned into a set of non-overlapping subnetworks, each subnetwork is associated with a COG-sublist.

Informally, a MAG-*list*, denoted by $\mathcal{M}$, contains a list of nodes where each node represents the node with maximum marginal gain in a COG-list. Note that the size of $\mathcal{M}$ is the number of non-overlapping subnetworks or components generated from the social network graph G. Figure 5 depicts an example of the structures of COG-sublists and MAG-list.

Definition 10 (MAG-*list*) Given the social network graph $G(V, E)$, let $\mathit{\Gamma} = \mathbf{Partition}(G)$ where $|\mathit{\Gamma}| = \ell$. Then, the

[2] glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

Algorithm 5: The *MAGConstruction* Algorithm.

```
Input: Non-overlapping subnetworks
       $\Gamma = \{G_0(V_0, E_0), G_1(V_1, E_1), \ldots, G_{\ell-1}(V_{\ell-1}, E_{\ell-1})\}$
       of the social network graph $G(V, E)$, the cascade
       influence function $\sigma(\cdot)$, the influence and conformity
       indices $(\Phi(v), \Omega(v))$ for all $v \in V$.
Output: MAG-list $\mathcal{M}$ and a set of COG-sublists of $\Gamma$ denoted by
       $\Upsilon$.
1  begin
2      initialize the MAG-list $\mathcal{M}$ of size $\ell$;
3      foreach $G_i(V_i, E_i) \in \Gamma$ do
4          initialize COG-sublist $\beta^i$;
5          foreach $v \in V_i$ do
6              $v.valid = 0$;
7              $\beta^i.append(v)$;
8          $\Upsilon.add(\beta^i)$;
9      for $iter = 1$ to $R$ do
10         for $i = 0$ to $\ell - 1$ do
11             compute $G_i'(V_i, E_i')$ by removing each edge $\overrightarrow{uv}$ from
               $G_i(V_i, E_i)$ with probability $1 - \Phi(u)\Omega(v)$;
12             foreach $v \in V_i$ do
13                 $v.gain += \sigma_i(v)$;
14     for $i = 0$ to $\ell - 1$ do
15         sort($\beta^i$) by $\beta^i.gain$ in descending order;
16         $top(\beta^i).valid = 1$;
17         $\mathcal{M}[i] = top(\beta^i)$;
18     return $(\mathcal{M}, \Upsilon)$
```

MAG**-list**, denoted by $\mathcal{M}$, is a list of nodes of size ℓ where $\mathcal{M}[i] = top(\beta^i)\, \forall\, 0 \leq i < \ell$.

To facilitate the discussions on algorithms, we assume some auxiliary functions of nodes. Given a node v, $append(v)$ and $remove(v)$ append and remove v from a node set or COG-sublist, respectively. Algorithm 5 outlines the MAG-list construction algorithm. For each subnetwork $G_i(V_i, E_i)$ it first initializes a COG-sublist β_i and populates it by setting the $valid$ attributes of the nodes to 0 (Lines 3–8). Next, for nodes in each subnetwork G_i, it computes the marginal gains based on the proposed cascade model and assigns them to the list of nodes in β_i (Lines 9–13). The nodes in β_i are sorted in descending order of their marginal gains (Line 15). We set the valid attributes of all $top(\beta^i)$ to 1 as in the first iteration their marginal gains equal to their influences (Line 16). Lastly, the algorithm constructs the MAG-list $\mathcal{M}$ by inserting the top element $top(\beta^i)$ of each β^i (Line 17). Note that the MAG-list construction requires only a linear traversal over the COG-sublists.

7 Seeds selection

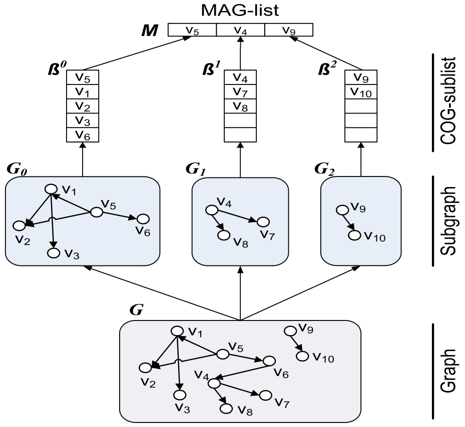


Fig. 5 The structures of MAG-list and COG-sublists

Algorithm 6: The *SeedsSelection* Algorithm.

```
Input: Graph $G(V, E)$, the budget $k$, the cascade influence
       function $\sigma(\cdot)$, the influence and conformity indices
       $(\Phi(v), \Omega(v))$ for all $v \in V$, MAG-list $\mathcal{M}$ and COG-sublist
       $\beta_i$ for $i = 0, \ldots, \ell - 1$
Output: Seed set $S$ of nodes, $|S| = k$
begin
    while $\sum_{i=0}^{\ell-1} |S_i| < k$ do
        $v' = \mathcal{M}[r] = \arg\max_{v \in \mathcal{M}} (v.gain)$;
        if $v'.valid == 1$ then
            $S_r.append(v')$;
            $V.remove(v')$;
            $V_i.remove(v')$;
            $\beta^r.remove(v')$;
        else
            update($\beta^r$, $G(V_r, E_r)$, $\sigma(\cdot)$, $\Phi(\cdot)$, $\Omega(\cdot)$) /* Alg. 7 */;
        $\mathcal{M}[r] = top(\beta^r)$;
    return $S = \bigcup_{i=0}^{\ell-1} S_i$;
```

Let us first illustrate the seeds selection phase intuitively with the example in Fig. 5. The MAG-list $\mathcal{M}$ contains the nodes v_5, v_4, and v_9. In the first round of iteration, we select the node having maximum marginal gain from the MAG-list (i.e., v_5) as a candidate. We check if its gain is up-to-date ($valid$ field is 1). Recall from Algorithm 5, the top node in each COG-sublist is marked as valid. That is, the flag values (valid) of other nodes in a COG-sublist are set to 0 (not up-to-date). Thus, v_5 is valid in this round. We insert it into S and remove it from G and the COG-sublist β^0. Now v_1 moves to the top of β^0 and hence it is copied to $\mathcal{M}[0]$. In the next round, assume that v_1 is the node with the maximum marginal gain in $\mathcal{M}$ and hence is selected as a candidate. However, v_1's gain is not up-to-date. Consequently, we need to update v_1's gain as it may change due to addition of v_5 in S. The update process works as follows. We recompute the marginal gain of v_1 in β_0 and check whether v_1's gain is still the highest. If it is, then we mark v_1 as valid. Otherwise, we move v_1 to the correct position in the COG-sublist β_0 to ensure that the list remains sorted in descending order. After the update is completed, we select the next candidate for the next round. The seed selection process terminates when there are k nodes in S.

Algorithm 7: The *update* Algorithm.

```
   Input: COG-sublist $\beta^r = [v_1, v_2, \dots, v_j]$, Subnetwork
          $G_r(V_r, E_r)$, the influence and conformity indices
          $(\Phi(v), \Omega(v))$ for all $v \in V$ and the cascade influence
          function $\sigma_r(\cdot)$.
   Output: Updated COG-sublist $\beta^r$ whose top node's
           $top(\beta^r).valid = 1$
1  begin
2      for $iter = 1$ to $R$ do
3          compute $G'_i(V_i, E'_i)$ by removing each edge $\overrightarrow{uv}$ from
           $G_i(V_i, E_i)$ with probability $1 - \Phi(u)\Omega(v)$;
4          $top(\beta^r).gain += \sigma_r(top(\beta^r))$
5      if $top(\beta^r).gain \geq \beta^r[1].gain$ then
6          $top(\beta^r).valid = 1$;
7      else
8          foreach $i = 1$ to $j - 1$ do
9              if $\beta^r[i-1].gain < \beta^r[i].gain$ then
10                 $t = \beta^r[i-1]$;
11                 $\beta^r[i-1] = \beta^r[i]$;
12                 $\beta^r[i] = t$;
13     return $\beta^r$;
```

Algorithm 6 outlines the aforementioned intuition for finding the seed set using the MAG-list. It iteratively selects from the MAG-list the node v' having the maximum gain as a candidate (Line 3). Then, the algorithm checks whether the v''s gain is updated by evaluating its *valid* field (Line 4). If it is already updated, then it inserts v' into S_r. Next, it removes v' from the COG-sublist β^r as well as G and continue to the next round (Lines 5–8). Otherwise, the candidate node's gain is not up-to-date. Consequently, the algorithm updates v''s marginal gain and reorders β^r by invoking the *update* procedure (Lines 10), which we shall elaborate later. Then, it updates $\mathcal{M}[r]$ using the top element $top(\beta^r)$ (Line 11). The algorithm terminates when there are k nodes in S.

On-demand update Algorithm 7 outlines the update strategy. In order to speed up seeds selection, we propose a strategy that dynamically updates a specific COG-sublist only when it is demanded. We refer to this strategy as *on-demand update*. Observe from Algorithm 6 only when a node is selected to be a candidate for S and its marginal gain is not up-to-date with respect to the current S, the update process is invoked for a specific COG-sublist β^r (Line 10 in Algorithm 6). Consequently, a node's marginal gain is not always guaranteed to be valid. Instead, it is updated only when demanded. The algorithm recomputes the marginal gain of $top(\beta^r)$ based on C^2 (resp. C^3) model (Lines 2–4 in Algorithm 7). Observe that we only need to recompute G'_r by random removing edges for R iteration when $v \in V_r$ is selected. In contrast, state-of-the-art greedy approaches [8] iteratively recompute it over the whole network G for R times after selecting a node into the seed set. That is, it takes $O(Rm)$ operations. Instead, as we have limited the update of the marginal gain to a subnetwork G_r, the time complexity for selecting a node improves to $O(Rm_r)$ (i.e., m_r is the number of edges in the subnetwork G_r).

Next, it checks whether $top(\beta^r)$ still achieves the highest marginal gain in β^r (Line 5). If it does, then the node's $valid$ field is set to 1 (Line 6). Otherwise, it reorders COG-sublist β^r by moving the top element toward the tail to a proper position j such that $\beta^r[j-1].gain > \beta^r[j].gain > \beta^r[j+1].gain$ (Lines 8–12). Observe that our reordering strategy (Lines 5–12) is similar to that of CELF [28]. Finally, the algorithm returns the COG-sublist β^r.

For example, consider the aforementioned scenario in Fig. 5. As discussed earlier, we have selected the first seed v_5. In the next round of seed selection, v_1 is the node with the highest marginal gain. As its gain is not up-to-date, v_1 and β^0 are updated. During the update, the gain of v_1 changes to 0 with respect to the seed $S = \{v_5\}$. Consequently, the algorithm reorders v_1 in β^0 to the tail as it has the least marginal gain.

The aforementioned update strategy makes sense in our partitioning-based IM problem as the gains of elements in a COG-sublist are not affected by other COG-sublists, which results from the fact that a node v is only connected with other nodes in v's COG-sublist. Thus, if v is considered to be selected for the seed set S then it will only affect the marginal gain of those nodes that belong to the same COG-sublist as v. The marginal gain of nodes in other COG-sublists is not affected and need not to be updated. Observe that in CINEMA only the global MAG-list and a *specific* COG-sublist are kept in the memory at an arbitrary timepoint. Hence, the memory required for CINEMA is only $O(n')$ where n' denotes the number of nodes in the largest component. Consequently, it is more efficient than several existing algorithms [8,23,28] which have $O(n)$ space complexity.

Synchronized update An alternative update strategy, which we refer to as *synchronized update*, guarantees that the nodes in MAG-list are all up-to-date. That is, in this strategy, we update all the gains of nodes in β^i whenever an update happens for β^i. Thus, in each iteration, $\mathcal{M}[i].valid$ is always guaranteed to be 1 and we can directly select the best node from $\mathcal{M}$ and update the corresponding COG-sublist β^i. For instance, reconsider the aforementioned example. Based on synchronized update strategy, we do not need to wait for checking v_1's $valid$ field. Instead, we update β^0 as soon as v_5 is inserted into S, guaranteeing that the nodes in the MAG-list are all valid. Although this strategy may avoid unnecessary selection of candidate nodes from $\mathcal{M}$, it introduces significant amount of updating and reordering of the COG-sublist. An empirical comparison of these two update strategies is reported in Appendix L.3 of ESM.

Theorem 5 *The time complexity of* CINEMA *is* $O(k'm'n' + kTRm')$ *where* k' *is the number of iterations in* CASINO *in Algorithm 1.*

Proof The proof is given in Appendix I of ESM. □

8 CINEMA on distributed platform

We now discuss how CINEMA can be elegantly adopted on a distributed and parallel environment. We utilize *Hadoop*, which has been widely used in the research of distributed computing [31].

We first partition a large social network graph into subnetworks and distribute each subnetwork to a process node (i.e., slave machine). In each subnetwork, there is a COG-sublist storing the expected influence for the nodes within the local subnetwork in descending order. The partitioning of the social network graph results in ℓ non-overlapping subnetworks. As a node in one subnetwork cannot influence nodes in another subnetwork, the marginal gain computation and reordering of nodes in different subnetworks can run in parallel. Ideally, we should distribute ℓ subnetworks to ℓ slave machines. However, due to the limitation of computing resources, we distribute the subnetworks into q slaves ($q < \ell$), each of which contains a set of subnetworks. Furthermore, we store the MAG-list in a master machine, and the COG-sublists are distributed into several slave machines. Each slave machine is in charge of recomputing the marginal gain in one or more subnetworks and outputs the updated top nodes. For ease of explanation, assume that a subgraph G_i is distributed to i-th slave machine and thus the map stage and the reduce stage can be defined as follows.

In map stage, each process node scans G_i, updates the corresponding COG-sublist and selects the node with the most expected influence in G_i according to Algorithm 7. Note that this computation is independent of other subgraphs and thus it can be run iteratively and continuously. Thus, the *map* function is defined for each node u in G_i. It issues an intermediate key/value pair $((u, \sigma_i(u)), (i, j))$ where j is the order of u in the COG-sublist. In reduce stage, each process node collects all values associated with an intermediate key to generate a list of candidate seeds, namely the MAG-list. Thus, the *reduce* function in each process node is defined for each intermediate key/value pair $((u, \sigma_i(u)), (i, j))$ and works as follows. It collects all available intermediate key/value pairs $((u', \sigma_{i'}(u')), (i', j'))$ for $i' \neq i$ and ranks them according to the expected influence (i.e., $\sigma_i(u)$ and $\sigma_{i'}(u')$), the result of which is denoted as r. Then, the output for reduce stage is key/value pairs $((u, \sigma_i(u)), r + j - 1)$. The combinations over all output for the reduce task constructs the final list of selected seeds (i.e., MAG-list). The pseudocode of the reduce stage is outlined in Appendix J of ESM.

Note that due to the aforementioned strategy, the selection of nodes from MAG-list is independent of the updating and reordering of COG-sublists. Moreover, the updating and reordering of COG-sublists are also independent of each other. Consequently, selection of nodes from the MAG-list as well as updating and reordering of a COG-sublist can be processed in parallel.

9 Performance study

CINEMA is implemented in Java. Note that there is no existing IM algorithm that is conformity-aware. Nevertheless, since our goal is to demonstrate that our proposed technique produces superior quality influence spread without sacrificing running time compared to existing greedy approaches, we confine ourselves to compare CINEMA against state-of-the-art IC model-based IM techniques [5–8,23]. For fair comparison, we implement all the algorithms in Java. We run all experiments on 3.2 GHz Quad-Core machines with 16GB RAM, running OS X Mountain Lion.

9.1 Experimental setup

We use four real-world context-free social network graphs for our experiments (see Appendix K of ESM for statistics). *Phy* and *Hep* are two academic collaboration networks from the paper lists in two different sections of the e-print *arXiv*. Each node in the network represents an author, and the number of edges between a pair of nodes is equal to the number of papers the two authors collaborated. The *Hep* network is from the "High Energy Physics - Theory" section with papers from 1991 to 2003. The *Phy* network represents the full paper list of the "Physics" section during 1991 and 2003.[3] Note that these datasets are also used in several prior studies such as [7,8,18,23,28]. The *Wiki-talk*[4] contains all the users and discussions in Wikipedia from its inception to January 2008. Nodes in the network represent Wikipedia users, and edges represent talk page editing relationship. Lastly, the *LiveJournal*[5] is an on-line community with almost 10 million members; a significant fraction of these members are highly active [24].

We use the *Twitter* dataset to investigate the performance on a context-aware network (see Appendix K of ESM for statistics). The dataset was crawled using the *Twitter* API[6] during Dec 2010 to Feb 2011. By invoking the *trends* module in the API, we extracted top-20 trends (keywords) at hourly duration. We retrieved up to 1,500 tweets for each trend by invok-

[3] Net and Phy are downloaded from http://research.microsoft.com/enus/people/weic/graphdata.zip.

[4] http://snap.stanford.edu/data/wiki-Talk.html.

[5] http://snap.stanford.edu/data/soc-livejournal1.html.

[6] http://dev.twitter.com/doc.

ing the *search* module.[7] Then, we identified the relationships between all tweets in the dataset. Note that we remove non-English tweets (using *Twitter* API). In order to compute accurate influence and conformity indices, we need to have large context-aware network. We removed spam trend keywords which contain only meaningless IDs. Thus, we selected top 492 trends that contain more than 1,000 tweets to compute the indices. For each trend (topic), we identified all tweets associated to it. Then, the edges connecting different tweets using the `'@'` tag are extracted and their signs are assigned as positive or negative. Additionally, there exists another tag `'RT'` in many tweets indicating that a tweet author supports another author's opinion by re-tweeting it. That is, if an author u directly re-tweets another tweet of v, then it indicates that u wants to distribute this tweet to her followers. Hence, we assign positive signs to such re-tweet edges.

We run the following algorithms.

- *Greedy*-IC*:* The general greedy algorithm [23] for the IC model.
- *MixGreedy*-[i]*:* The *MixGreedy* algorithms [8] for the IC and WC models where i is "IC" or "WC", respectively.
- *SingleDiscount:* The single discount heuristic [8] algorithm.
- *DegreeDiscount*-IC*:* The degree discount heuristic [8] for the IC model. Also, we create a variant by adapting the propagation probability from user-specified parameter p to $\Phi(u)\Omega(v)$ (denoted by *DegreeDiscount*-c^2).
- MIA-N: The MIA-N [6] algorithm (with $q = 0.9$) that can be applied to IC-N model. Additionally, we use a variant of MIA-N by adapting the propagation probability from IC-N to C^2 model as above (denoted by MIA- N- C^2).
- PMIA: The PMIA algorithm [7] for the IC model. We also create a variant (denoted by PMIA- C^2) that is adapted to our C^2 model by modifying the propagation probability of each path to $\Phi(u)\Omega(v)$.
- TIC: The TIC algorithm [5] for the topic-aware IC model.[8]
- CINEMA-[i]: The CINEMA algorithm for the C^2, C^3, and IC^2 models where $i \in \{C^2, C^3, IC^2\}$.
- D-CINEMA-C^2-*q:* The distributed CINEMA algorithm with q slaves.

We set $T = 5$ (number of iterations in gain computation under WC model and C^2/C^3 model) and $R = 20{,}000$ (number of rounds of simulation) for all the models, which is in line with the experiments in [6,8]. We vary k from 10 to 100 for different seed set size. Note that we exclude [17] as it requires historical action logs, which is not available from the data sources.

9.2 Experimental results

We now investigate the performance of CINEMA from a variety of aspects and report some of the results here. Additional results are in Appendix L of ESM.

Indices distribution We first investigate the performance of the CASINO algorithm in computing the influence and conformity indices. Figure 6 depicts the distribution heatmap of influence index versus conformity index for *Twitter* and *LiveJournal* datasets. The heatmaps of other datasets are given in [29]. For each individual u in a network, we compute her influence index $\Phi(u)$ and conformity index $\Omega(u)$ and represent it as a point in the influence-conformity 2D plane. Then we separate the plane into grids of size 0.005×0.005 and count the number of points in each grid. The color shade of a grid denotes the number of points residing in it. Note that both influence index and conformity index are normalized into the range of $[0, 1)$. For each figure, we explicitly draw a boundary line along which the vertices exhibit identical influence index and conformity index. Observe that the line separates the influence-conformity plane into two areas. In the sequel, we refer to the top area as '*Area I*' and the down one as '*Area II*'. The points belonging to '*Area I*' exhibit higher influence index compared to conformity index, indicating that individuals in this area are more prone to influence others than being influenced. We refer to them as *influence-biased*. On the other hand, the points in '*Area II*' represent individuals who are conforming in nature. That is, they are more prone to be influenced than influencing others. We refer to these individuals as *conformity-biased*.

Figure 6a, b show the distributions of influence and conformity indices for the top-2 topics (`Mumford & Sons` and `BornThisWayFriday`) with the most number of tweets (4,390 and 4,046, resp.). Observe that 40 and 45 % of all the individuals fall in '*Area I*' for Fig. 6a, b, respectively. Notably, both these figures exhibit *certain* influence-biased characteristics. That is, there are a few influence-biased individuals who exhibit very high influence but are not easily influenced by others. This may be due to the fact that *Twitter* is driven by user conversations where majority are commenting or following a few individuals who started the conversations. Figure 6c plots the distribution of indices computed over *all* topics. In this case, 41 % of all individuals belong to '*Area I*'. The most influential author has influence index of 0.138, whereas the most conforming individual has a conformity index of 0.082. Similar phenomenon is also observed in *LiveJournal* dataset (Fig. 6d). A case study of influence-biased and conformity-biased users in *Twitter* dataset is given in Appendix L.1 of ESM.

Effect of partitioning algorithms Recall that in the first phase of CINEMA, we may employ a ℓ-way partitioning algorithm

[7] Given a user-specified keyword (trend), this module returns up to 15 result pages, each containing 100 tweets.

[8] We thank the first author for sending us the code.

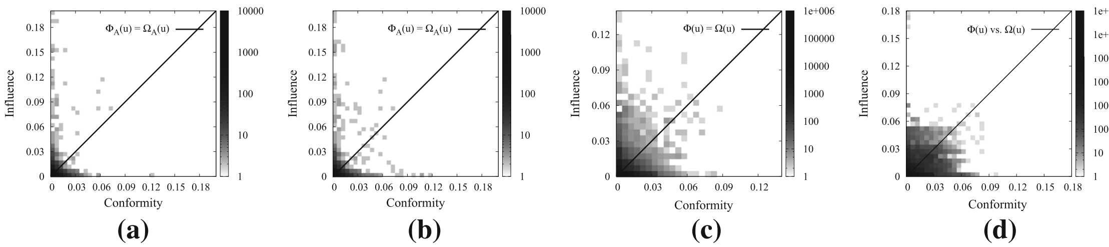


Fig. 6 Influence versus Conformity Distribution Heatmap of *Twitter* and *LiveJournal*. **a** `Mumford & Sons`, **b** `BornThisWayFriday`, **c** Twitter (all topics), **d** LiveJournal

in CLUTO to create the subnetworks. We now empirically justify the reason for choosing CLUTO as the graph partitioning algorithm for CINEMA. Specifically, we compare three partition algorithms, namely CLUTO, *EdgeBetweenness* [15] and SCOTCH [35], on *Wiki-talk* and *LiveJournal* and investigate their effects on influence spreads. *EdgeBetweenness* method partitions a given graph by removing a specified number of edges that exhibit the highest betweenness score. Both CLUTO and SCOTCH are multi-level algorithms aiming to partition a graph into clusters by removing a limited number of edges. Both CLUTO and SCOTCH partition *LiveJournal* in around 1,000 s, whereas *EdgeBetweenness* takes more than 100 h.

Figure 7a, b show the influence spreads (the number of influenced nodes) generated by feeding the partitioned graphs from these three algorithms into the remaining three phases of CINEMA (Algorithm 4). Observe that the seeds quality is not affected significantly by adopting different partitioning methods. Hence, a key advantage of CINEMA is that it is not tightly coupled to any specific partitioning technique. In the sequel, CLUTO is used to partition the networks as it is faster than *EdgeBetweenness*. Note that adoption of a more superior partitioning technique which is particularly sensitive to the power-law characteristics of real-world networks will only enhance the influence spread quality of CINEMA. Also, the increase of ℓ means that more edges are ignored resulting in poorer performance of CINEMA. Note that in practical applications the seed set tends to be small due to budget restriction.

We also show in Fig. 7a, the effect on the influence spread if we employ CINEMA on the entire *Wiki* network *without* partitioning it (e.g., $\ell = 1$). Observe that the influence spread quality is affected modestly due to partitioning (decreased by less than 4.2 % for $\ell = 40$).

Seeds selection of C^2 *and* IC^2 *models* Consider our proposed C^2 and IC^2 models. In this set of experiments, we compare the seeds selection results by using these models to test whether our proposed models are consistent with the theories proposed in social psychology. We run CINEMA under C^2 and IC^2 models from which a pair of seeds sets are generated, namely S^2 and IS^2, respectively. We conduct a series of analysis over both sets by varying k (i.e., number of seeds selected).

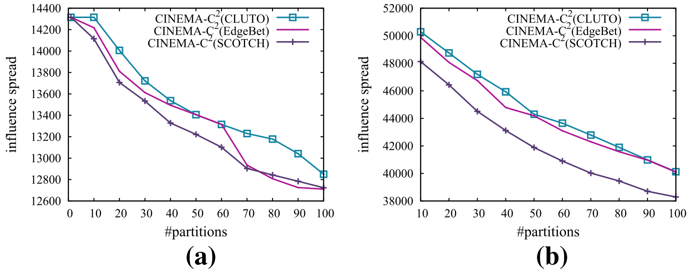
Table 3 Pearson correlation coefficient between $rank(v, S^2)$ and $rank(v, IS^2)$

	Hep	Phy	Wiki-talk
$k = 10$	1	1	0.987345
$k = 20$	0.913142	0.96391	0.927575
$k = 30$	0.961885	0.982364	0.905673
$k = 40$	0.926156	0.982319	0.916321
$k = 50$	0.952166	0.984696	0.942389

Let $rank(v, S)$ denote the ranked order of node v in S. For instance, $rank(v, S^2)$ represents the order of v to be selected by CINEMA-C^2. Firstly, we compute the Pearson correlation coefficient of $rank(v, S^2)$ and $rank(v, IS^2)$ of the same seed v for different k values (i.e., $rank(v, S^2) \leq k$). Table 3 reports the results. Clearly, $rank(v, S^2)$ and $rank(v, IS^2)$ are highly correlated with each other. Secondly, we plot the pair of ranked orders for these nodes in a 2D plane. Figure 8 shows the ranked distribution of all the selected seeds for S^2 and IS^2 with $k = 50$. X-axis represents the rank of a seed in S^2 while Y-axis represents that in IS^2. For instance, a seed u with $rank(v, S^2) = 3$ and $rank(v, IS^2) = 4$ is plotted at the position (3, 4). Moreover, we fit all the 50 points to the line $rank(v, S^2) = rank(v, IS^2)$ and examined the parameters for this hypothesis. In summary, data points in *Hep*, *Phy* and *Wiki-talk* fit to $rank(v, IS^2)$ with extremely high confidence with $p \ll 0.05$. This indicates that for the representative online networks, nodes that are more than one hop away (e.g., friend-of-friend) do not significantly impact the seeds selection. Consequently, the social forces principle degenerates to $F = f(SN)$ for the IM problem for these networks. In other words, C^2 is a good approximation for IC^2. In the subsequent experiments, we shall use the C^2 model instead of the IC^2 model (Fig. 7).

Influence spread In this set of experiments, we compare the influence spreads of CINEMA against various approaches. A pertinent issue is how do we compare it among different techniques under different cascade models? Simply comparing the expected influence spreads between different cascade models can be misleading. For instance, assume the seed sets computed using *MixGreedy*-IC and *MixGreedy*-WC are S_1 and S_2, respectively. Let the expected influence spread

Fig. 7 Effect of partitioning algorithms. **a** Wiki, **b** LiveJournal



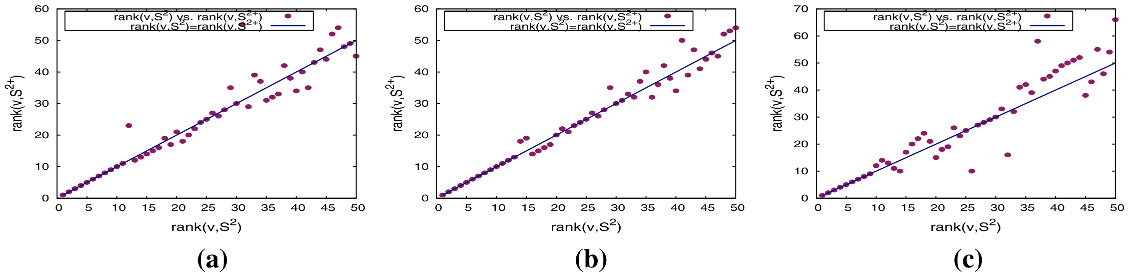


Fig. 8 Comparison of the seeds generated by CINEMA-C^2 and CINEMA-IC^2. **a** *Hep* ($R^2 = 0.98$, $F = 2{,}008.07$, $p = 1.97 \times 10^{-41}$), **b** *Phy* ($R^2 = 0.99$, $F = 6{,}335.31$, $p = 1.74 \times 10^{-53}$), **c** *Wiki-talk* ($R^2 = 0.97$, $F = 1{,}430.73$, $p = 6.34 \times 10^{-38}$)

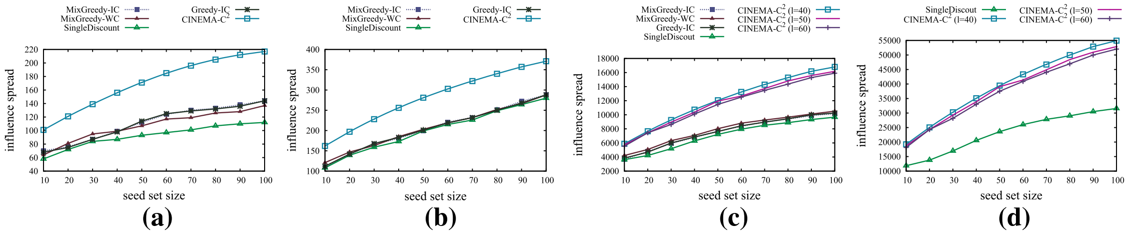


Fig. 9 [Best viewed in color] Influence spread. **a** Hep, **b** Phy, **c** Wiki-talk, **d** LiveJournal (color figure online)

of S_1 under IC model and S_2 under WC model be E_1 and E_2, respectively. Clearly, simply comparing E_1 and E_2 will not shed light on which algorithm is better in terms of influence spread. We address this issue along two dimensions. First, we adopt a strategy similar to [9] by *unifying* the cascade model under which the expected influence is computed. Specifically, we utilize C^2 model instead of IC and WC models and compare the spreads generated by CINEMA against conformity-unaware algorithms. Second, we adapt some of the existing models (PMIA [7], MIA-N [6], *DegreeDiscount*) to C^2 model by modifying the propagation probability from user-specified p to $\Phi(u)\Omega(v)$ as mentioned earlier.

We select k (varies from 10 to 100) nodes using different approaches and compute the expected influence of those nodes. Parameter p in *MixGreedy*-IC and *MixGreedy*-WC is set to 0.01 which is in line with [8]. In fact, the value of p does not affect the final seed set selected according to the study in [8]. Note that TIC [5] is a supervised method requiring learning of topic distributions from action logs which is only available in *Twitter* dataset. Hence, its performance is reported later in the context of context-aware networks.

Figures 9 and 10 report the performances of different approaches. We can make the following observations. First, the CINEMA-C^2 curves follow diminishing pattern which support the submodular nature of influence function. Second, it consistently performs better than conformity-unaware approaches. We attribute it to the design of CINEMA tailored specifically to the C^2 model. Interestingly, the number of influenced users estimated by CINEMA is counter-intuitive to the definition of “activation probability” in CINEMA (Recall from Definition 5). This is primarily due to the following reason. There are a number of nodes, although relatively small, whose influence or conformity indices are more than 0.1 (both accounts for over 1 % of all nodes). Moreover, 5 % of all nodes in each dataset have either influence index or conformity index more than 0.05. The limited number of seeds selected by IM algorithms tends to belong to these groups of nodes. Furthermore, the neighbors as well as the neighbors of neighbors of these seeds tend to belong to these groups as well. Hence, the final activated number of users is not as small as expected.

Third, CINEMA is more scalable than state-of-the-art IM approaches. *MixGreedy*-IC, *MixGreedy*-WC, *Greedy*-IC, PMIA, PMIA- C^2, MIA-N, and MIA- N- C^2 did not finish execution in *LiveJournal* dataset due to excessive memory usage. Additionally, PMIA, PMIA-C^2, MIA-N, and MIA-N-C^2 did not

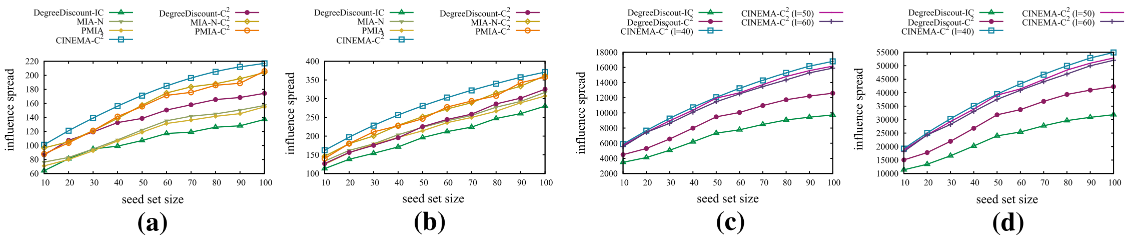
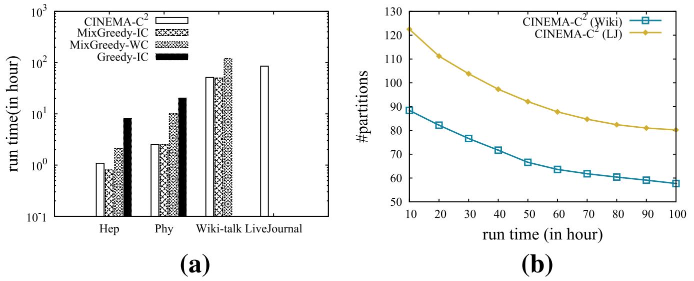


Fig. 10 [Best viewed in color] Influence spread (contd.). **a** Hep, **b** Phy, **c** Wiki-talk, **d** LiveJournal (color figure online)

Fig. 11 Running times. **a** Running times, **b** Effect of ℓ



finish execution for the *Wiki-talk* dataset as well. In contrast, CINEMA does not consume so much memory and can finish execution due to the usage of MAG-list and COG-sublist.

Fourth, CINEMA-C^2 outperforms the conformity-ujnaware heuristic-based approaches (PMIA, MIA-N, *SingleDiscount*, *DegreeDiscount*) consistently for all networks. Although these approaches are shown to be orders of magnitude faster than greedy approaches [8], the influence spreads computed by these approaches can be as low as 42 % and 38 % of the size of influence spread computed by CINEMA-C^2 for the *Hep* and *LiveJournal* datasets, respectively. In addition to ignoring conformity of nodes in computation of influence probabilities, these heuristics approaches either discount a node's degree if it has a neighbor selected as a seed or eliminating paths whose propagation probability is less than some specified thresholds. However, discounting the degree does not incorporate the fact that most highest-degree nodes are clustered and hence it cannot avoid unnecessary targeting. Moreover, to the best of our knowledge, there is no evidence showing that for a single seed, higher degree will definitely lead to larger scope of influence spread. Nodes with lower degree may influence more users. Importantly, as discussed in Sect. 1, we believe that the seed set quality is paramount to companies as they would like to maximize the influence spreads of their products. Hence, *it cannot be significantly compromised*. Fifth, the quality of influence spread of the aforementioned heuristic approaches can be improved significantly by adapting them to incorporate our proposed C^2 model. Specifically, the performances of *DegreeDiscount*-C^2, PMIA-C^2, and MIA-N-C^2 (Fig. 10a, b) are superior to their conformity-unaware counterparts highlighting the benefits of the C^2 model. They also account for as much as 90 % of the influence spread of CINEMA.

Lastly, Figs. 9c, d and 10c, d depict the influence spread of CINEMA-C^2 on *Wiki-talk* and *LiveJournal* networks for different values of ℓ. Recall that both networks were partitioned using ℓ-way partitioning algorithm which may results in removal of some edges. As ℓ increases the size of each subnetwork may decrease. Consequently, more edges are ignored resulting in slightly lower quality of seeds. In spite of this, CINEMA-C^2 shows superior performance compared to the conformity-unaware techniques.

Running times We now investigate the response times of various approaches. For the *LiveJournal* and *Wiki-talk* datasets, the response times of CINEMA-C^2 *include* initial partitioning attempt using the BFS technique. Figure 11a reports the running times (the running times of individual phases of CINEMA are reported in Appendix L.2 of ESM). Observe that in spite of the additional steps of network partitioning and indices computation, the running times of CINEMA-C^2 are almost the same with *MixGreedy*-IC and much less than *MixGreedy*-WC and *Greedy*-IC in both *Hep* and *Phy* (*MixGreedy*-IC and *MixGreedy*-WC cannot finish in *LiveJournal* due to excessive memory usage. Similarly, *Greedy*-IC cannot finish in *Wiki-talk* and *LiveJournal*.). Thus, it is reasonable to be applied in real applications. Since it is already demonstrated in [8] that the heuristic-based techniques under IC and WC are orders of magnitude faster than all greedy algorithms, for the sake of visual clarity, we do not plot them here. However, the gain in speed is achieved by sacrificing quality of influence spreads as reported earlier. Lastly, we study the effect of varying ℓ on the running time of CINEMA-C^2. Figure 11b depicts that the running time of CINEMA-C^2 over the *LiveJournal* and *Wiki-talk* networks. Observe that the running time decreases as ℓ increases.

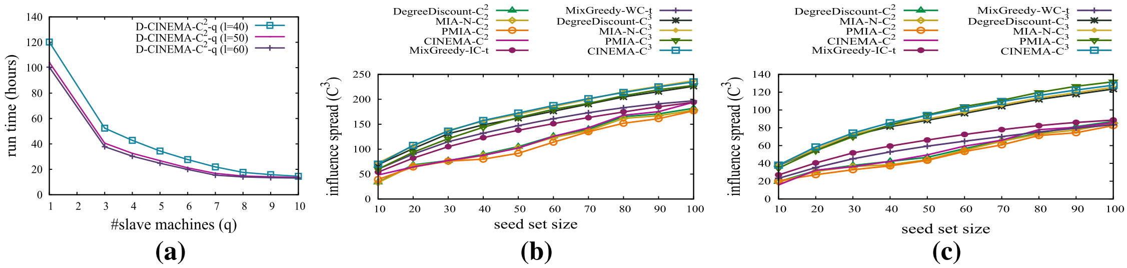


Fig. 12 [Best viewed in color] Distributed CINEMA and influence spread on random-sampled topics in *Twitter*. **a** `D-CINEMA` for $k = 50$, **b** `Topic1:PolandNeedsTakeMeHomeTour`, **c** `Topic2:WhenJustinSmileBeliebersSmile` (color figure online)

CINEMA *on distributed platform* Lastly, we test the scalability of D-CINEMA over *LiveJournal* in distributed platform by varying the number of slaves q. The influence spread results remain stable when q ranges from 1 to 10. Hence, varying q does not affect the influence spread results as long as $q \leq \ell$. Figure 12a plots the running times over different q values. Obviously, as q increases the running time decreases sublinearly indicating that adding more slaves will speedup the algorithm. Due to the limitation of computation resource, we can only test the running time with $q \leq 10$. Observe that CINEMA on distributed platform takes less than 6 h when $q = 10$ in comparison to CINEMA on a single machine (over 80 h). In summary, D-CINEMA- c^2 is able to complete the maximization task much faster than CINEMA while preserving similar influence spread.

CINEMA *on context-aware networks* Next, we study the C^3 model and influence spread in a context-aware network (*Twitter*). Recall that if the network is context-aware then we compute the influence and conformity indices of nodes with respect to a given topic $\mathbb{T}$ in each partitioned subgraphs. Thus, a pair of topic-specific influence and conformity indices is associated with each node. Then, we can maximize the influence with respect to a given topic $\mathbb{T}$ by selecting k seeds using Algorithm 4. We also create context-aware variants of *MixGreedy* (denoted by *MixGreedy*-IC/WC-`t`), MIA-N-C^2, PMIA-C^2 and *DegreeDiscount*-C^2 (denoted by MIA-N-C^3, PMIA-C^3 and *DegreeDiscount*-C^3) where the topic-specific subgraph related to topic $\mathbb{T}$ is first extracted from the social network graph and then *MixGreedy* (resp. *DegreeDiscount*-C^2, MIA-N-C^2, PMIA-C^2) is executed on the subgraph. Figure 13a–h plot the influence spreads for the top 4 topics with maximum number of tweets (`Mumford & Sons`, `BornThisWayFriday`, `WeLoveTokioHotel` and `Mubarak`).

Clearly, CINEMA-C^3 consistently outperforms all existing methods as well as CINEMA-C^2. This is because the context-unaware approaches are unable to distinguish whether a seed exhibits any influence with respect to a specific topic $\mathbb{T}$. Many of the seeds selected by these approaches may not influence anyone else for topic $\mathbb{T}$. Thus, the influence spreads from these seeds are poor. Observe that CINEMA is superior to C^2-adapted versions of MIA-N, PMIA and *DegreeDiscount* algorithms. Similarly, topic-sensitive approaches such as TIC did not outperform it. In contrast to CINEMA-C^3, TIC assumes a topic distribution for each item and IM is performed nodes which probably will tweet on a specific topic but with less probability on influence propagation. A comparative study of specific seeds selected by different models is given in Appendix L.4 of ESM.

Figure 14a–h plot the running times of the benchmark techniques for the top 4 topics. Observe that the context-aware approaches (i.e., CINEMA-C^3, *MixGreedy*-IC/WC-`t`, MIA-N-C^3, PMIA-C^3 and *DegreeDiscount*-C^3) spend less than an hour to select seeds for each topic. In contrast, the context-free greedy approaches (i.e., *MixGreedy*-IC/WC and CINEMA-C^2) spend around 10 h to select seeds. Interestingly, although TIC is context-aware and a heuristic-based approach, it requires more than 10 h for learning the topic distribution and node authorities in the *Twitter* dataset.

Lastly, since the behavior of top trends in *Twitter* can be qualitatively different from its entirety, we test our model on two randomly selected topics.[9] There are 410 and 204 users for these two topics. The influence spread results are shown in Fig. 12b, c. Observe that context-aware techniques are still superior to context-free ones although the gap is more modest (compared to Fig. 13). Moreover, the performances of context-aware heuristics (e.g., MIA-N-C^3, PMIA-C^3) and CINEMA-C^3 are similar. This is primarily due to the small size of topic-aware subnetworks, making it difficult to distinguish the impact of conformity and non-conformity approaches.

9.3 Summary

From the experimental results, we find the following: (a) IM techniques that incorporate the interplay of conformity and influence produce superior quality seeds compared to state-of-the-art conformity-unaware techniques. Additionally, in [29], we have demonstrated how such technique can

[9] Randomly selected from a twitter corpus which random-sampled 21,410,330 tweets during Jan 22 to 29, 2013.

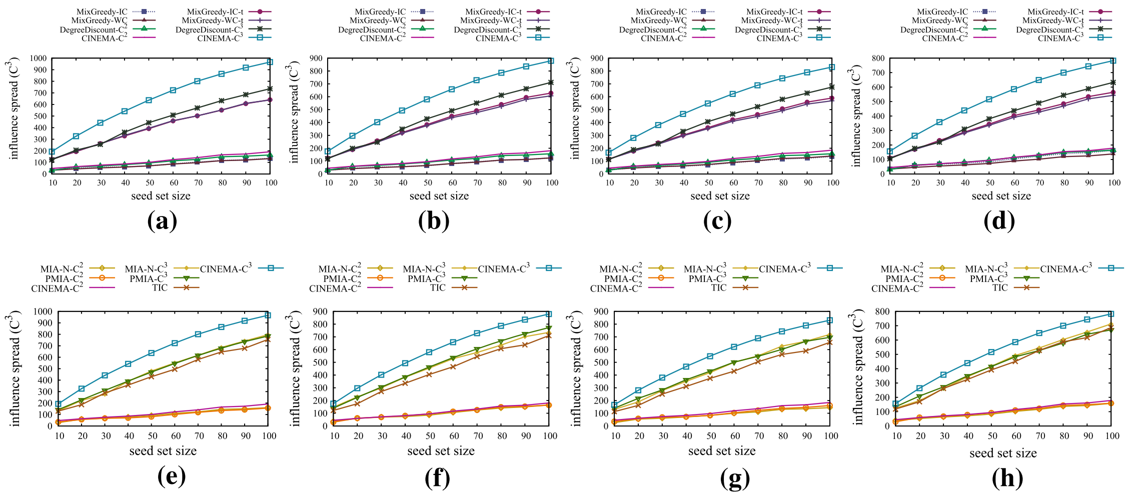


Fig. 13 [Best viewed in color] Influence spread on a context-aware network. **a–d** Comparison with *MixGreedy* and *DegreeDiscount*-IC; **e–h** Comparison with MIA-N, PMIA, and TIC. **a** Mumford & Sons. **b** BornThisWayFriday. **c** WeLoveTokioHotel. **d** Mubarak. **e** Mumford & Sons. **f** BornThisWayFriday. **g** WeLoveTokioHotel. **h** Mubarak (color figure online)

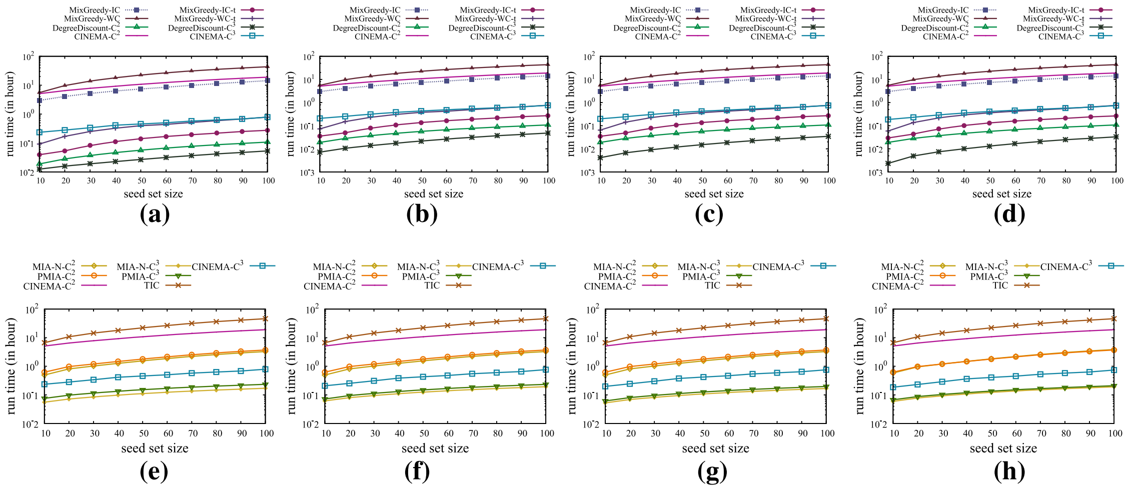


Fig. 14 [Best viewed in color] Running times on a context-aware network. **a–d** Comparison with *MixGreedy* and *DegreeDiscount*-IC; **e–h** Comparison with MIA-N, PMIA, and TIC. **a** Mumford & Sons. **b** BornThisWayFriday. **c** WeLoveTokioHotel. **d** Mubarak. **e** Mumford & Sons. **f** BornThisWayFriday. **g** WeLoveTokioHotel. **h** Mubarak (color figure online)

also be used for link prediction (summarized in Appendix L.5 of ESM). (b) Partitioning a social network into pieces of non-overlapping subnetworks and distributing the influence and conformity computation to these components is a viable strategy for computing high-quality seeds at lower computational cost. Particularly, the choice of a partitioning technique has limited impact on the quality of the influence spread. (c) The distributed variant of CINEMA further reduces the response time by an order of magnitude without compromising on the spread quality. (d) The context-aware variant of CINEMA produces superior quality seeds within reasonable time compared to context-unaware strategies by limiting the IM problem to portions of the network that are relevant to a specific topic of interest. (e) The proposed C^2 (resp. C^3) model is a good approximation of the IC^2 (resp. IC^3) model.

10 Conclusions

In this paper, we propose a novel conformity-aware greedy algorithm called CINEMA to address the influence maximization (IM) problem. Specifically, it is built on top of a novel conformity-aware cascade model that incorporates the interplay between conformity and influence to estimate influence probabilities. CINEMA first partitions the network into a set of subnetworks and for each of these subnetworks it obtains the influence probabilities of nodes from the underlying network by computing both influence as well as conformity indices of nodes. Then, each subnetwork is associated with a COG-sublist which stores the marginal gains of its nodes in descending order. The node with maximum marginal gain in each COG-sublist is stored in a structure called MAG-list. CINEMA exploits these lists along with an on-demand update strategy for marginal gains to efficiently find the seed set. We also demonstrate that CINEMA can be elegantly realized on a *MapReduce* framework by maintaining the MAG-list in a central machine and the maximization of influence for the subnetworks are distributed into several machines and computed in parallel. Lastly, CINEMA can be seamlessly extended to support context-specific influence maximization by targeting "topic-relevant" individuals in a social network. Our empirical study has demonstrated that CINEMA and its context-aware and distributed variants have excellent real-world performance compared to state-of-the-art IM approaches.

Acknowledgments Hui Li and Jiangtao Cui are supported by National Nature Science Foundation of China (under NSFC Grant Nos. 61202179 and 61173089), SRF for ROCS, SEM and the Fundamental Research Funds.

References

1. A-lias: Lingpipe 4.0.1. http://alias-i.com/lingpipe
2. Aronson, E., Wilson, T.D., Akert, R.M.: Social Psychology, 5th edn. Prentice Hall, Englewood Cliffs NJ (2004)
3. Asch, S.: Opinions and social pressure. Sci. Am. **193**, 31–35 (1955)
4. Asch, S.E.: Effects of group pressure upon the modification and distortion of judgment In: Guetzkow, H. (ed.) Groups, leadership and men, pp. 177–190. Carnegie Press, Pittsburgh, PA (1951)
5. Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. In: IEEE ICDM (2012)
6. Chen, W., Collins, A., et al.: Influence maximization in social networks when negative opinions may emerge and propagate. In: SDM (2011)
7. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: ACM KDD (2010)
8. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: ACM KDD (2009)
9. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: IEEE ICDM (2010)
10. Cialdini, R.B., Goldstein, N.J.: Social influence: Compliance and conformity. Annu. Rev. Psychol. **55**, 591–621 (2004)
11. Crano, W.D.: Effects of sex, response order, and expertise in conformity: a dispositional approach. Sociometry **33**, 239–252 (1970)
12. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: OSDI (2004)
13. Edelson, M., Sharot, T., Dolan, R.J., Dudai, Y.: Following the crowd: brain substrates of long-term memory conformity. Science **333**(6038), 108–111 (2011)
14. Epley, N., Gilovich, T.: Just going along: nonconscious priming and conformity to social pressure. J. Exp. Soc. Psychol. **35**(6), 578–589 (1999)
15. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS **99**, 7821–7826 (2002)
16. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: ACM WSDM (2010)
17. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: A data-based approach to social influence maximization. In: PVLDB **5**(1) (2011)
18. Goyal, A., Lu, W., Lakshmanan, L.V.S.: Simpath: An efficient algorithm for influence maximization under the linear threshold model. In: IEEE ICDM (2011)
19. James, K., Zollman, S.: Social Structure and the Effects of Conformity. Springer, Berlin (2008)
20. Jiang, Q., Song, G., et al.: Simulated annealing based influence maximization in social networks. In: AAAI (2011)
21. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. In: ACM WSDM (2011)
22. Jung, K., Heo, W., Chen, W.: IRIE: Scalable and robust influence maximization in social networks. In: IEEE ICDM (2012)
23. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: ACM KDD (2003)
24. Kim, J., Kim, S.K., Yu, H.: Scalable and parallelizable processing of influence maximization for large-scale social networks. In: IEEE ICDE (2013)
25. Klucharev, V., Munneke, M.A.M., et al.: Downregulation of the posterior medial frontal cortex prevents social conformity. J. Neurosci. **31**(33), 11934–11940 (2011)
26. Latane, B.: The psychology of social impact. Am. Psychol. **36**(4), 343–356 (1981)
27. Leskovec, J., Huttenlocher, D.P., Kleinberg, J.M.: Predicting positive and negative links in online social networks. In: ACM WWW (2010)
28. Leskovec, J., Krause, A., et al.: Cost-effective outbreak detection in networks. In: ACM KDD (2007)
29. Li, H., Bhowmick, S.S., Sun, A.: Casino: towards conformity-aware social influence analysis in online social networks. In: ACM CIKM (2011)
30. Li, H., Bhowmick, S.S., Sun, A.: Cinema: conformity-aware greedy algorithm for influence maximization in online social networks. In: ACM EDBT (2013)
31. Li, F., Ooi, B.C., Ozsu, T., Wu, S.: Distributed data management using MapReduce. ACM Comp. Surv. **46**(3), 31:1–31:42 (2014)
32. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: ACM CIKM (2009)
33. Liu, L., Tang, J., et al.: Mining Topic-level Influence in Heterogeneous Networks. In: CIKM (2010)
34. Pedro, D., Matt, R.: Mining the network value of customers. In: ACM KDD (2001)
35. Pellegrini, F., Roman, J.: Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In: HPCN. Springer, Berlin (1996)
36. Pornpitakpan, C.: The persuasiveness of source credibility: a critical review of five decades' evidence. J. Appl. Soc. Psychol. **34**(2), 243–281 (2004)
37. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: ACM KDD (2002)
38. Schloegel, K., Karypis, G., Kumar, V.: Parallel static and dynamic multi-constraint graph partitioning. Concurr. Comput. Pract. Exp. **14**(3), 219–240 (2002)
39. Stricklanda, B.R., Crowne, D.P.: Conformity under conditions of simulated group pressure as a function of the need for social approval. J. Soc. Psychol. **58**(1), 171–181 (1962)
40. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: ACM KDD (2009)
41. Tang, J., Wu, S., Sun, J.: Confluence: Conformity influence in large social networks. In: ACM KDD (2013)

42. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: ACM KDD (2010)
43. Yu, X., Liu, Y., et al.: A quality-aware model for sales prediction using reviews. In: ACM WWW (2010)