CINEMA: Conformity-Aware Greedy Algorithm for Influence Maximization in Online Social Networks*

Hui Li
School of Computer Science and Technology
Xidian University, China
hli@xidian.edu.cn

Sourav S Bhowmick
School of Computer Engineering
Nanyang Technological University, Singapore
assourav@ntu.edu.sg

Aixin Sun
School of Computer Engineering
Nanyang Technological University, Singapore
axsun@ntu.edu.sg

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 propose a novel greedy algorithm called CINEMA that generates high quality seed set by exploiting this model. 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. 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. Our empirical study with real-world social networks demonstrates that CINEMA generates superior quality seed set compared to state-of-the-art IM approaches.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

*This work was done when the first author was pursuing doctoral degree in Nanyang Technological University.

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

General Terms

Algorithms, Experimentation, Performance, Reliability

Keywords

Social networks, Influence maximization, Conformity, Network partitioning, Greedy algorithm

1. INTRODUCTION

With the emergence of large-scale online social networking applications (SNA), we are now faced with the opportunity to analyze social network data at unprecedented levels of scale and temporal resolution for a wide variety of applications. However, translating the research techniques of traditional SNA to these large-scale online data-intensive applications is a daunting task. In this paper, we present our work towards addressing one of the challenges, namely the *influence maximization* problem.

Given a social network as well as an *influence propagation* (or *cascade*) model, the problem of *influence maximization* (IM) is to find the 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 [16]. Domingos and Richardson [21, 23] are the first to study influence maximization as an algorithmic problem. Kempe et al. [16] 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 cascade models, namely the *independent cascade* (IC) model, the *weighted cascade* (WC) model, and the *linear threshold* (LT) model (see Section 3.1 for details). A key strength of the proposed 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 [7]. Recently, several greedy approaches [7, 18, 26] 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.

- The aforementioned greedy approaches still take days to find seeds in real-world networks containing millions of nodes [12]. To alleviate the performance bottleneck, several heuristic-based techniques [4–7, 13] have been proposed which are orders of magnitude faster than the greedy approaches. However, despite the blazing speed of these heuristics-based techniques, greedy approaches are more reliable as the former often produces inferior-quality seed set (detailed in Section 2).

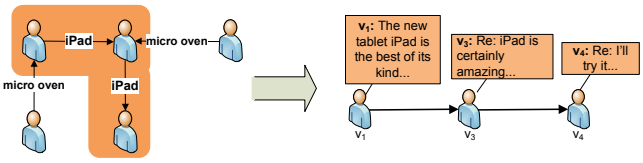


Figure 1: An example of real-world social network.

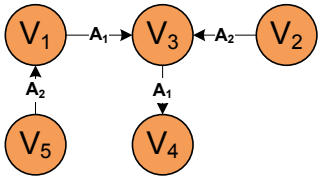


Figure 2: Graph representation of Figure 1.

Note that seed set quality is of great importance 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 [1–3]. It is well known that humans will readily conform to the wishes or beliefs of others [1, 2]. It was perhaps a surprise when Solomon Asch [2, 3] 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 [2, 3, 8, 10, 14, 25] and more recently in neuroscience [9, 17], to the best of our knowledge, it has not been investigated in the context of online IM problem.

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 to this question affirmatively using an example. Consider Figure 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 Figure 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 *iPad* in 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 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 the following [16].

$$\sigma(v) = \sum_X Prob[X] \cdot \sigma^X(v) \quad (1)$$

In the above equation, $Prob[X]$ denotes the probability that all the edges in X are activated. For instance, the expected influence size of v_3 under 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 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 $\Phi(.)$) and *conformity* (denoted by $\Omega(.)$) *values* of all nodes, respectively, computed using the technique described in [19]. Specifically, these values are computed by analyzing the sentiments expressed by the edges in the underlying network (detailed in Section 2). 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.*

Model	$\sigma(v_1)$	$\sigma(v_2)$	$\sigma(v_3)$	$\sigma(v_4)$	$\sigma(v_5)$
IC ($p(\overrightarrow{uv}) = 0.5$)	1.75	1.75	1.5	1	**1.875**
WC ($p(\overrightarrow{uv}) = 1/d(v)$)	1.67	1.67	**2**	1	1.83
c^2 ($p(\overrightarrow{uv}) = \Phi_1(u)\Omega_1(v)$)	**1.73**	1	1.49	1	1

Table 1: Expected influence size of nodes in Figure 2.

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

Table 2: Nodes' influence and conformity indices.

1.2 Overview & Contributions

In this paper, we propose a novel greedy algorithm called CINEMA (**C**onformity-aware **IN**flu**E**nce **MA**ximization) that solves the IM problem in real-world social networks by efficiently utilizing the interplay of conformity and influence. It is based on a novel *conformity-aware cascade model* (c^2 model) to study the influence propagation process by taking into account the conformity behavior of nodes in a social network. Specifically, in this model influence

Symbol	*Definition*
$G(V, E)$	A social network graph
n	number of vertices in G
m	number of edges in G
$G_i(V_i, E_i)$	i^{th} component (subnetwork) in $G(V, E)$
Γ	A set of subnetworks (components)
m'	$\max_{E_i \in E} \lvert E_i \rvert$
k	number of seeds to be selected
ℓ	number of connected components (subnetworks)
R	number of rounds of simulation
β^i	A cog-sublist
Υ	A set of cog-sublists
$\mathcal{M}$	mag-list
S	seed set
S_i	seed nodes selected from $G_i(V_i, E_i)$
$\Omega(\cdot)$	conformity index
$\Phi(\cdot)$	influence index
$\overrightarrow{uv}$	the edge pointing from u to v
$\sigma_i(\cdot)$	influence function under cascade model C_i
T	number of iterations in gain computation

Table 3: Key notations used in this paper.

propagation of an edge is modeled as a product of its influence and conformity.

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. A node's influence in one component is not significantly affected by nodes in other components as many large social networks are comprised of series of communities and clusters where a piece of information can easily spread within the community but hard to propagate from one to another [11]. Consequently, each node's influence computation and updates can be limited to the component it resides. Next, for each of these subnetworks, cinema computes the *influence* and *conformity indices* of nodes using a recently proposed algorithm called casino [19].

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, 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. 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. It is not necessary to update all nodes in the mag-list.

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

- To the best of our knowledge, this is the first im approach that leverages conformity of nodes to generate superior quality seeds. The approach 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 interplay of influence and conformity of nodes.
- We present a simple but effective data structure called mag-*list* which facilitates efficient computation of im problem. It also provides an efficient framework to support updates of nodes' influences.
- Departing from 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 although cinema exploits existing graph partitioning and conformity computation techniques, our solution ensures that it 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.
- By applying cinema to real social networks of various sizes, 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 Section 2 we review related work. In Section 3 we present the *conformity-aware cascade model* (c^2), while in Section 4 we present an overview of our im solution cinema based on this model. In Sections 5 and 6 we discuss in detail the mag list construction and seed selection steps of cinema, respectively. Section 7 presents our experimental evaluation and, lastly, Section 8 offers our conclusions. The notations used in this paper are given in Table 3.

2. RELATED WORK

2.1 Greedy IM Approaches

Domingos et al. [21] 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. [16] 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 $f(A_{greedy}) \geq (1 - 1/e) \max_{|A| \leq k} f(A)$ [20]. That is, a greedy algorithm 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 influence maximization problem, which is computationally very expensive for real-world social networks.

Leskovec et al. [18] proposed an algorithm called celf (Cost-Effective Lazy Forward) that is reported to be 700 times faster than the algorithm 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. [7] 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. They demonstrated that the running time of *MixGreedy* is smaller than CELF.

Wang et al. [26] 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. They developed a cost function that optimized 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.

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. [7] 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 [4] 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 [7]. 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. The LDAG model [6] is similar to PMIA except that it is specifically designed for the *linear threshold* model. More recently, Goyal et al. [13] proposed a heuristic-based approach called SIMPATH in order to improve the seed quality of LDAG by consuming less memory. However, *they demonstrated that* CELF *still outperforms the aforementioned heuristic-based approaches with respect to quality of influence spread.* Specifically, unlike greedy algorithms, the quality of influence spread of these models are not guaranteed to be within 63% of the optimal.

Jiang et al. [15] 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 afterwards, 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% to 8%).

Chen et al. [5] proposed a model called IC-N (IC model with negative opinions) 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. As we shall see later, CINEMA addresses this problem by computing a pair of *influence* and *conformity* indices for each individual.

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 [26]) 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 [26], 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 Historical Log-based Approach

Recently, Goyal et al. [12] proposed a *credit distribution (*CD*) model* that leverages on historical *action logs* of a network to learn how influence flows in the network and use this to estimate influence spread. 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. Based on this assumption the CD model 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. In contrast to our approach, this model suffers from t-

wo key limitations. Firstly, it depends on the availability of 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, similar to existing work, the CD model does not incorporate conformity information in computing influenceability.

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 [1–3]. 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 [1–3, 8, 10, 14, 25]. We are inspired by these conformity studies and utilize it for influence spread computation in online IM problem.

Recently, we proposed an algorithm called CASINO to study the interplay between influence and conformity of each individuals in online social networks [19]. Specifically, it computes the *influence* and *conformity indices* of each individual in the network. The intuition behind the conformity index computation is that each edge in the network represents positive or negative attitudes of individuals toward opinions of others. However, 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*). On the other hand, links in many *context-free* networks (*e.g., Slashdot*, *Epinions*) are explicitly labeled with signs. Hence, CASINO assigns signs to the edges by analyzing the sentiment expressed by the edge. Specifically, for each edge $\overrightarrow{uv}$, it identifies 5-level sentiment (*i.e.,* like, somewhat like, neutral, somewhat dislike, dislike) expressed at both ends (*LingPipe* [27] is used for implementing this). If the sentiments at both ends are similar (*sentiment similarity threshold* is less than ε), then the edge is denoted as positive. Otherwise, it is a negative. Hence, each network containing both positive and negative edges can be represented using a pair of graphs $G^+(V, E^+)$ and $G^-(V, E^-)$ denoting the induced graph of positive edges E^+ (trust/agreement) and negative edges E^- (distrust/disagreement), respectively. Finally, given the signed network, CASINO iteratively compute the influence (denoted by $\Phi(.)$) and conformity indices (denoted by $\Omega(.)$) of each individual using the following equations.

$$\Phi(v) = \sum_{\overrightarrow{uv} \in E_i^+} \Omega(u) - \sum_{\overrightarrow{uv} \in E_i^-} \Omega(u) \quad (2)$$

$$\Omega(u) = \sum_{\overrightarrow{uv} \in E_i^+} \Phi(v) - \sum_{\overrightarrow{uv} \in E_i^-} \Phi(v) \quad (3)$$

where $\Omega(u)$ and $\Phi(v)$ represent the conformity and influence indices of nodes u and v, respectively.

In contrast, in this work we go beyond computation of conformity index of a node. Specifically, CINEMA leverages on the conformity of nodes to address the IM problem. As we shall see later, such strategy enables us to obtain superior quality influence spread.

3. CONFORMITY-AWARE CASCADE MODEL

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 and state-of-the-art cascade models that have been considered in the literature.

3.1 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. We use n and m to denote the numbers of nodes and edges, respectively. The influence maximization (IM) problem is defined as follows [16].

DEFINITION 1. *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:*

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

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 following cascade models as defined in [16].

- *Independent cascade (IC) model.* Let A_i be the set of nodes that are influenced in the i-th round and $A_o = |S|$. For any $(u, v) \in E$ 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 an independent probability p, which is referred to as the *propagation probability*. Thus, if there are t neighbors of v that are in A_i, then $v \in A_{i+1}$ with probability $1-(1-p)^t$. This process is repeated until A_{i+1} is empty.
- *Weighted cascade (*WC*) model.* The WC model can be considered as an instance of IC model [16]. Let $(u, v) \in E$. In this model, if u is influenced in round i, then v is influenced by u in round $(i + 1)$ with probability $1/v.degree$. Thus, if v has t neighbors influenced at the i-th round then the probability for a node v to be influenced in the next round is $1 - (1 - 1/v.degree)^t$.
- *Linear threshold (LT) model.* In this model, each node v has a threshold θ_v uniformly and randomly chosen from 0 to 1; this represents the weighted fraction of v's neighbors that must become influenced (active) in order for v to be influenced. All nodes that were influenced in step $(i - 1)$ remains so in step i, and any node v is influenced when the total weight of its influenced neighbors is at least θ_v.

The optimum solution to the IM problem is NP-hard for the aforementioned cascade models [16]. However, as discussed in the preceding section, 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 [16].

DEFINITION 2. *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 towards 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 [16].

3.2 Conformity-Aware Cascade Model (C^2 Model)

In the preceding section, we demonstrated 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, which can be computed from the underlying social network using a state-of-the-art conformity computation technique (*e.g.,* [19])[1]. Inspired by this finding, we define the *conformity-aware cascade* (c^2) model as follows.

Definition 3. *Let A_i be the set of nodes that are influenced in the i-th round and $A_0 = S$. For any $(u, v) \in E$ 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:*

$$1 - \prod_{u \in A_i, (u,v) \in E} (1 - \Phi(u)\Omega(v))$$

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

Theorem 1. *Given a social network graph $G(V, E)$, the influence function $\sigma(\cdot)$ under c^2 model is submodular.*

Proof. Let S_1 and S_2 be two sets of nodes such that $S_1 \subseteq S_2$. $R(v, X)$ denotes the set of all nodes that can be reached from v on all the activated edges that are in X. Consider the expression of $\sigma^X(S_1 \cup \{v\}) - \sigma^X(S_1)$. It denotes the number of elements in $R(v, X)$ that are not already in $\bigcup_{u \in S_1} R(u, X)$, which is at least as large as the number of elements in $R(v, X)$ that are not in $\bigcup_{u \in S_2} R(u, X)$. That is $\sigma^X(S_1 \cup \{v\}) - \sigma^X(S_1) \geq \sigma^X(S_2 \cup \{v\}) - \sigma^X(S_2)$, which means that the function $\sigma^X(\cdot)$ is submodular. Moreover, we have shown that $\sigma(\cdot)$ can be computed from $\sigma^X(\cdot)$ using Equation 1. It means $\sigma(\cdot)$ is a non-negative linear combination of another submodular function $\sigma^X(\cdot)$. Hence $\sigma(\cdot)$ is also submodular. □

4. OVERVIEW OF CINEMA

In this section, we first formally define the partitioning-based influence maximization problem that is proposed in this paper. Then, we give an overview of key steps of the Algorithm CINEMA.

4.1 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 [7, 16, 18]. 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 4. *Given a budget k and a social network $G(V, E)$, let $\Gamma =$ **Partition**(G) be the partitions of G containing a set of subnetworks where $V = V_1 \cup V_2 \cup \ldots \cup V_{|\Gamma|}$, $V_i \cap V_j = \emptyset \; \forall \; i \neq j$, $0 \leq i, j < |\Gamma|$, 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 (*e.g., c^2, IC, WC*). 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 same cascade model (*e.g.,* c^2 model) to reflect many real-world applications.

Theorem 2. *Given the social network graph $G(V, E)$ and $\Gamma =$ **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 4 is also submodular.*

Proof. *(Sketch)* According to Definition 4, $\sigma(S)$ can be represented as the following.

$$\sigma(S) = \max_{\sum |S_i| = k} \sum \sigma_i(S_i)$$

Assume $S' \subset S, v \in V_t \setminus S_t$ where $t \in \{1 \ldots \ell\}$ and $S = S_1 \cup S_2 \cup \ldots \cup S_\ell, S' = S'_1 \cup S'_2 \cup \ldots \cup S'_\ell$, then $S'_i \subseteq S_i$. Besides, the following expression holds as $S_i \cap S_j = \emptyset \; \forall \; 0 < (i, j) \leq \ell$.

$$\begin{aligned} \sigma(S \cup \{v\}) - \sigma(S) &= \sigma_t(S_t \cup \{v\}) - \sigma_t(S_t) \\ \sigma(S' \cup \{v\}) - \sigma(S') &= \sigma_t(S'_t \cup \{v\}) - \sigma_t(S'_t) \end{aligned}$$

As $S'_t \subseteq S'$ and the influence function $\sigma_t(\cdot)$ is submodular, then $\sigma_t(S_t \cup \{v\}) - \sigma_t(S_t) \leq \sigma_t(S'_t \cup \{v\}) - \sigma_t(S'_t)$ holds according to the definition of submodularity. Thus, $\sigma(S \cup \{v\}) - \sigma(S) \leq \sigma(S' \cup \{v\}) - \sigma(S')$ holds too, which means that the influence function $\sigma(S)$ is submodular. □

Observe that the above theorem 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 [7].

4.2 Algorithm CINEMA

The CINEMA algorithm is outlined in Algorithm 1 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

[1] While we use [19] to compute conformity values of nodes, our proposed model and algorithm are not tightly coupled to this specific approach. Any other superior conformity computation technique can easily be leveraged by our proposed technique. However, this is orthogonal to the focus of this paper.

Algorithm 1: 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     Γ ← NetworkPartition(G);
3     foreach G_i ∈ Γ do
4         (G_i, (Φ_i(·), Ω_i(·))) ← ComputeConformity(G_i) /* Based
          on [19] */;
5     (M, ϒ) ← MAGConstruction(Γ, σ(·), Φ_i(·), Ω_i(·));
6     S ← SeedsSelection(G, k, σ(·), M, ϒ, Φ_i(·), Ω_i(·));
7 end
```

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 [24]. 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., Wiki-talk*) 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 (`glaros.dtc.umn.edu/gkhome/cluto/cluto/overview`) [24]. In Section 7, 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 Section 7 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 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 c^2 model. In this paper, we invoke the CASINO algorithm [19] for each subnetwork to achieve this goal (see Section 2.4). 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? 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 Section 5.

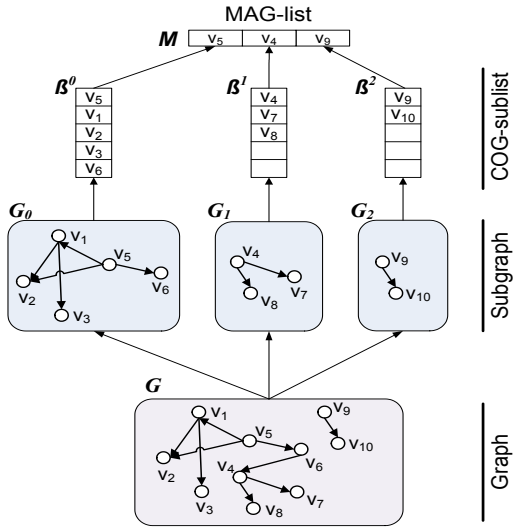


Figure 3: The structures of MAG-list and COG-sublists.

Phase 4: The seeds selection phase. Lastly, this phase exploits the MAG-list to compute the seed set S from the set of subnetworks (see Section 6). 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.

5. 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 ***co**mponent **g**ain sublist* (COG-sublist) which we shall be using to define MAG-list. Given a subnetwork $G_i(V_i, E_i)$ where $G_i \in \Gamma$, 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 3 depicts an example of the structures of COG-sublists and MAG-list.

Algorithm 2: 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 Γ denoted by Υ.

1 **begin**
2 initialize the MAG-list $\mathcal{M}$ of size ℓ;
3 **foreach** $G_i(V_i, E_i) \in \Gamma$ **do**
4 initialize COG-sublist β^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(β^i) by $\beta^i.gain$ in descending order;
16 $top(\beta^i).valid = 1$;
17 $\mathcal{M}[i] = top(\beta^i)$;
18 return $(\mathcal{M}, \Upsilon)$
19 **end**

DEFINITION 5. *Given the social network graph $G(V, E)$, let $\Gamma =$ **Partition**(G) where $|\Gamma| = \ell$. Then, the* **MAG-list***, denoted by $\mathcal{M}$, is a list of nodes of size ℓ where $\mathcal{M}[i] = top(\beta^i)\ \forall\ 0 \le i < \ell$.*

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 2 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.

6. SEEDS SELECTION

Let us first illustrate the seeds selection phase intuitively with the example in Figure 3. 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 2, the top node in each COG-sublist is marked as valid. That is, the marginal gains of all nodes except the top node in a COG-sublist is set to 0 (not up-to-date). Thus, v_5 is valid in this round. Consequently, 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.

Algorithm 3: 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 β_i for $i = 0, \ldots, \ell - 1$
Output: Seed set S of nodes, $|S| = k$

1 **begin**
2 **while** $\sum_{i=0}^{\ell-1} |S_i| < k$ **do**
3 $v' = \mathcal{M}[r] = \arg\max_{v \in \mathcal{M}} (v.gain)$;
4 **if** $v'.valid == 1$ **then**
5 $S_r.append(v')$;
6 $V.remove(v')$;
7 $V_i.remove(v')$;
8 $\beta^r.remove(v')$;
9 **else**
10 **update**(β^r,$G(V_r, E_r)$,$\sigma(\cdot)$, $\Phi(\cdot)$, $\Omega(\cdot)$) /* Algorithm 4 */;
11 $\mathcal{M}[r] = top(\beta^r)$;
12 return $S = \bigcup_{i=0}^{\ell-1} S_i$;
13 **end**

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 3 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.

6.1 On-demand Update

Algorithm 4 outlines the update strategy of CINEMA. 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 3 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 3). 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 model (Lines 2-4 in Algorithm 4). 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 [7] 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 towards the tail to a proper position j such that

Algorithm 4: The *update* Algorithm.

Input: cog-sublist $\beta^r = [v_1, v_2, \ldots, v_j]$, 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 β^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 uv from G_i(V_i, E_i)
           with probability 1 − Φ(u)Ω(v);
4          top(β^r).gain+ = σ_r(top(β^r))
5      if top(β^r).gain ≥ β^r[1].gain then
6          top(β^r).valid = 1;
7      else
8          foreach i = 1 to j − 1 do
9              if β^r[i − 1].gain < β^r[i].gain then
10                 t = β^r[i − 1];
11                 β^r[i − 1] = β^r[i];
12                 β^r[i] = t;
13     return β^r;
14 end
```

$\beta^r[j-1].gain > \beta^r[j].gain > \beta^r[j+1].gain$ (Lines 8-12). Observe that our reordering strategy (Lines 5-12) is similar to that of CELF [18]. Finally, the algorithm returns the cog-sublist β^r.

For example, consider the aforementioned scenario in Figure 3. 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 are 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 [7, 16, 18] which have $O(n)$ space complexity.

6.2 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. In the next section, we shall empirically investigate these two update strategies.

THEOREM 3. *The time complexity of* CINEMA *is* $O(k'm'n' + kTRm')$ *where* k' *is the number of iterations in* CASINO *[19].*

network	*nodes*	*edges*	*components*	m'
Phy	37,154	231,584	3,883	134,358
Hep	15,233	58,891	1,781	19,630
Wiki-talk	2,394,385	5,021,410	34	5,018,445

Table 4: Description of real-world networks.

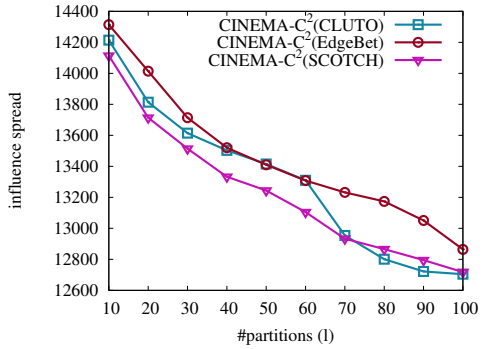


Figure 4: Partitioning algorithms.

PROOF. *(Sketch)* The time complexity of the indices computation step using CASINO (Line 2 in Algorithm 1) is $O(k'm'n')$ where k' is the number of iterations in influence and conformity indices computation [19]. The time complexity of the influence maximization step (Lines 4-6 in Algorithm 1) is $O(kTRm')$. Hence, the time complexity of CINEMA is $O(k'm'n' + kTRm')$. □

7. 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 IM techniques [5–7, 13]. Since several of these techniques are implemented in C++, for fair comparison we re-implement them in Java. We run all experiments on 1.86GHz Due-Core Intel 6300 machines with 4GB RAM, running Windows XP.

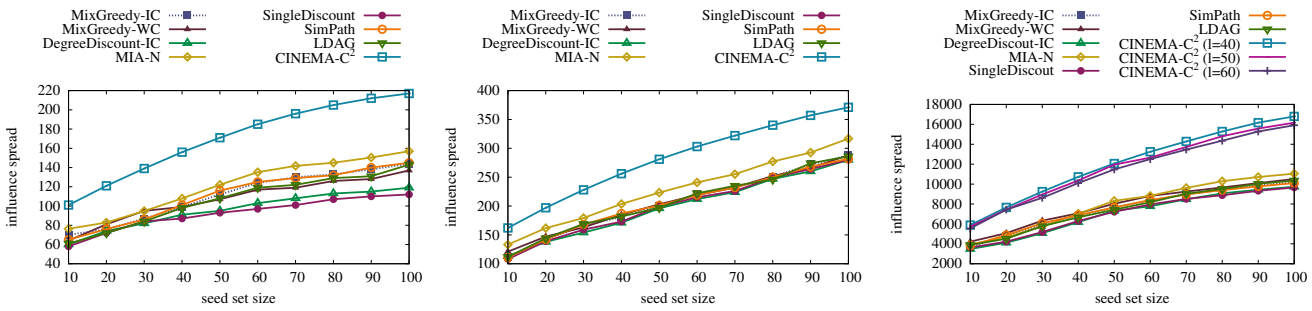
7.1 Experimental Setup

Table 4 summarizes the three real-world social network graphs used in our experiments. *Phy* and *Hep* are two academic collaboration networks from the paper lists in two different section 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[2]. Note that these datasets are also used in several prior studies such as [4, 7, 13, 16, 18]. The *Wiki-talk*[3] is a large network containing millions of nodes representing 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.

We run the following algorithms under different cascade models.

- *MixGreedy*-IC*:* The *MixGreedy* algorithm [7] for the IC model.

[2]Net and Phy are downloaded from `http://research.microsoft.com/enus/people/weic/graphdata.zip`.
[3]Downloaded from `http://snap.stanford.edu/data/wiki-Talk.html`.



(a) Spread of Hep (b) Spread of Phy (c) Spread of Wiki-talk

Figure 5: Influence spread.

- *MixGreedy*-wc*:* The *MixGreedy* algorithm for the wc model.
- *DegreeDiscount*-ic*:* The degree discount heuristic [7] for the ic model.
- *SingleDiscount:* The single discount heuristic [7] that can be applied to ic and wc models.
- mia-n*:* The mia-n heuristic [5] algorithm (with $q = 0.9$) that can be applied to ic-n model.
- ldag*:* The ldag algorithm [6] for the lt model.
- *SimPath:* The heuristic algorithm [13] for the lt model.
- cinema-c^2*:* The cinema algorithm for the c^2 model.

We set $T = 5$ (number of iterations in gain computation under wc model and c^2 model) and $R = 20000$ (number of rounds of simulation) for all the models, which is in line with the experiments in [5, 7]. We vary k from 10 to 100 for different seed set size. Note that we do not compare cinema with [12] as the latter requires historical action logs, which is not available from the data sources.

7.2 Experimental Results

We now investigate the performance of cinema from a variety of aspects.

7.2.1 *Effect of Partitioning Algorithms*

We first empirically justify the reason for choosing cluto as the graph partitioning algorithm for cinema. Specifically, we compare three partition algorithms, namely cluto, *EdgeBetweenness* [11] and scotch [22], on *Wiki-talk* network 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 *Wiki-talk* in around 100 seconds whereas *EdgeBetweenness* takes more than 20 hours.

Figure 4 shows the influence spreads (the number of influenced nodes) generated by feeding the partitioned graphs from these three algorithms into the last three phases in Algorithm 1. Observe that the seeds quality is not affected significantly by these three partitioning methods. Hence, a key advantage of cinema is that it is not necessary to be 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 than cluto 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.

7.2.2 *Influence Spread*

In this set of experiments, we compare the influence spreads of cinema against various approaches. However, 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 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. To address this issue, Chen et al. [6] adopts the strategy to *unify* the cascade model under which the expected influence is computed. That is, the results from *MixGreedy*-ic and *MixGreedy*-wc are all applied in lt model to test their performance in this model. We also adopt the same strategy. Specifically, we utilize c^2 model instead of ic and wc models and compare the spreads generated by cinema against conformity-unaware algorithms.

We select k (vary from 10 to 100) nodes using different approaches in the three networks and compute the expected influence of those nodes under c^2 model. Parameter p in *MixGreedy*-ic and *MixGreedy*-wc is set to 0.1 which is in line with [7]. In fact, the value of p does not affect the final seed set selected according to the study [7]. We use default settings for [6, 13] under lt model where the influence parameter θ in ldag is set to $1/320$ and pruning threshold η in *SimPath* is set to 10^{-3}. Besides, we set the negative opinion propagation factor q in mia-n algorithm as 0.9 which is in line with [5].

Figure 5 reports the performances of different approaches. We can make the following observations. Firstly, the cinema-c^2 curves follow diminishing pattern which support the submodular nature of influence function. Secondly, it consistently performs better than conformity-unaware approaches. We attribute it to the design of cinema tailored specifically to the c^2 model. Thirdly, Figure 5(c) depicts the influence spread of cinema-c^2 on the *Wiki-talk* network for different values of ℓ. Recall that *Wiki-talk* was 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.

Lastly, cinema-c^2 outperforms the heuristic-based approaches consistently for all networks. Although these approaches are shown to be orders of magnitude faster than greedy approaches [7], the influence spreads computed by these approaches can be as low as 42% and 40% of the size of influence spread computed by cinema-c^2 for the *Hep* and *Wiki-talk* datasets, respectively. In addition to ignoring conformity of nodes in computation of influence proba-

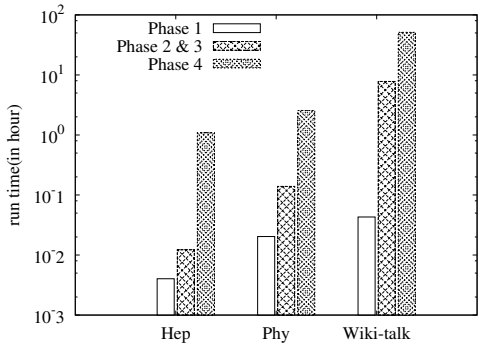


Figure 6: Cost of Phases 1-4.

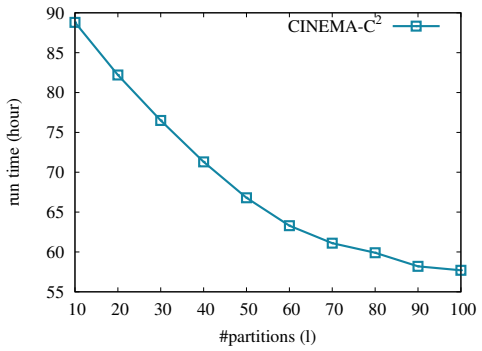


Figure 8: Effect of ℓ (Wiki-talk).

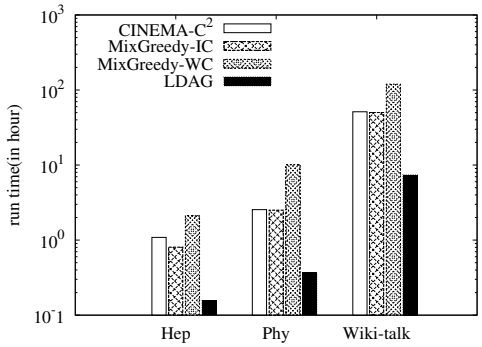


Figure 7: Running Times.

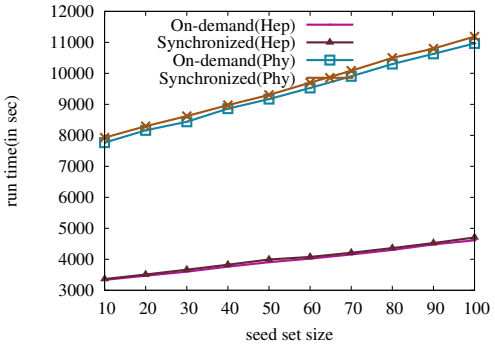


Figure 9: Update strategies (*Hep* and *Phy*).

bilities, these heuristics approaches discount a node's degree if it has a neighbor selected as a seed. However, discounting the degree does not incorporate the fact that most highest-degree nodes are clustered and hence it cannot avoid unnecessary targeting. Moreover, a node's influence may not always be reflected by its degree in that a node may influence another node over multiple hops while its degree only counts the nodes within a single hop. Importantly, as discussed in Section 1, we believe that the seed set quality is paramount to companies as they would like to maximize the influence spreads of their new products. Hence, *it cannot be significantly compromised*.

7.2.3 Cost of Phases 1-4

Next, we analyze the cost of Phases 1–4 of CINEMA. Figure 6 compares the running times of these phases for the four datasets. Since the running time of MAG-list construction is significantly smaller than the rest, we plot the total running time of Phases 1 and 2. Observe that the seed selection phase dominates the running time agreeing with our analysis in Section 4.2. Note that in order to ensure fair comparison with *Hep* and *Phy*, for *Wiki-talk* we depict only the partitioning time of the ℓ-way partitioning algorithm and not its initial failed attempt to partition using BFS technique.

7.2.4 Running Times

We now investigate the response times of various approaches. For the *Wiki-talk* dataset, the response times of CINEMA-C^2 *includes* initial partitioning attempt using the BFS technique. Figure 7 reports the running times of different approaches. Observe that in spite of the additional steps of network partitioning and indices computation, the running times of CINEMA-C^2 is almost the same with *MixGreedy*-IC and much less than *MixGreedy*-WC. Thus, it is reasonable to be applied in real applications. Since it is already demonstrated in [7] 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. Similar phenomenon exists in other heuristic-based techniques under LT model (*i.e.,* LDAG). Observe that LDAG runs 8 times faster than CINEMA. However, the gain in speed is achieved by sacrificing quality of influence spreads as reported in Figure 5. Lastly, we study the effect of varying ℓ results on the running time of CINEMA-C^2. Figure 8 depicts that the running time of CINEMA-C^2 over the *Wiki-talk* network. Observe that the running time decreases as ℓ increases.

7.2.5 On-demand vs. Synchronized Update

Lastly, we compare the on-demand and synchronized update strategies introduced earlier and justify our choice of the former. Note that the choice of using one of these strategy only affects the update performance of MAG-list and COG-sublist and not the seed set quality. Figures 9 and 10 plot the comparison of the running times between the two strategies for different values of k. The running times of both strategies increase linearly with k. Besides, the on-demand strategy is slightly better than the synchronized one which also agrees with our discussion in the preceding section.

8. CONCLUSIONS & FUTURE WORK

The influence maximization (IM) problem for online social networks focuses on finding the set of k users (seeds) so that they eventually influence the largest number of individuals (influence spread) in the network. We propose a novel conformity-aware greedy algorithm called CINEMA to address the IM problem. It first partitions the network into a set of subnetworks and for each of these subnetworks, in contrast to existing approaches, it obtains the influence probabilities of nodes from the underlying network by computing both influence as well as conformity indices of nodes. Then,

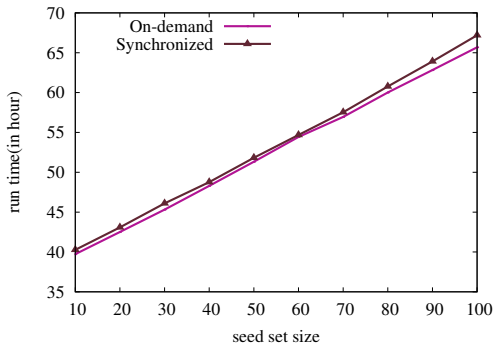


Figure 10: Update strategies (Wiki-talk).

each subnetwork is 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 structure called MAG-list. CINEMA exploits these lists along with an on-demand update strategy for marginal gains to efficiently find the seed set. Our empirical study has demonstrated that CINEMA has excellent real-world performance compared to state-of-the-art IM approaches. Specifically, we demonstrated that partitioning-based, conformity-aware IM strategy is a more realistic solution as it can not only improve computation time but also maintain high quality seed set that is more relevant to real-world applications. We also advocated that despite the blazing speed of heuristics-based techniques, greedy approaches are more reliable as the former may produce inferior-quality seed set. Note that seed set quality is of great importance to companies as they would like to maximize the influence spreads in order to reach largest number of potential customers.

It is not difficult to realize that the partitioning-based strategy of CINEMA paves way for its easy adoption on a distributed platform. Specifically, the MAG-list can be maintained in a central machine and the maximization of influence for the subnetworks are distributed into several machines and computed in parallel. As for future work, we plan to investigate such distributed and parallel strategy. In summary, the results of this paper are an important first step in this regard.

Acknowledgement: Hui Li was supported by NSFC (No. 61202179 and 61173089).

9. REFERENCES

[1] E. Aronson, T. D. Wilson, and R. M. Akert, *Social Psychology*, 5th ed. Prentice Hall, Feb. 2004.

[2] S. E. Asch, "Effects of group pressure upon the modification and distortion of judgment," In *H. Guetzkow (ed.) Groups, leadership and men*, Pittsburgh, PA: Carnegie Press, 1951.

[3] S.E. Asch, "Opinions and social pressure," *Scientific American*, 193, 1955.

[4] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM KDD*, 2010.

[5] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincón, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate," in *SDM*, 2011.

[6] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *IEEE ICDM*, 2010.

[7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *ACM KDD*, 2009.

[8] R. B. Cialdini and N. J. Goldstein, "Social Influence: Compliance and Conformity", *Annual Review of Psychology*, Vol 55, 2003.

[9] M. Edelson, T. Sharot, R. J. Dolan, Y. Dudai, "Following the Crowd: Brain Substrates of Long-Term Memory Conformity", *Science*, 333(6038), 2011.

[10] N. Epley and T. Gilovich, "Just Going Along: Nonconscious Priming and Conformity to Social Pressure." *Journal of Experimental Social Psychology*, 35(6), 1999.

[11] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *PNAS*, vol. 99, no. 12, June 2002.

[12] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *PVLDB*, vol. 5, no. 1, 2011.

[13] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *IEEE ICDM*, 2011.

[14] K. James and S. Zollman, "Social structure and the effects of conformity", Springer-Verlag, 2008.

[15] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *AAAI*, 2011.

[16] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *ACM KDD*, 2003.

[17] V. Klucharev, M. A. M. Munneke, A. Smidts, G. Fernandez, "Downregulation of the Posterior Medial Frontal Cortex Prevents Social Conformity", *The Journal of Neuroscience*, 31(33), 2011.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *ACM KDD*, 2007.

[19] H. Li, S. S. Bhowmick, and A. Sun, "Casino: towards conformity-aware social influence analysis in online social networks," in *ACM CIKM*, 2011.

[20] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions. i,," *Math. Programming*, 14(3), 1978.

[21] D. Pedro and R. Matt, "Mining the network value of customers," in *ACM KDD*, 2001.

[22] F. Pellegrini and J. Roman, "Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs," in *HPCN*. Brussels, Belgium. LNCS 1067: Springer, April 1996.

[23] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *ACM KDD*, 2002.

[24] K. Schloegel, G. Karypis, and V. Kumar, "Parallel static and dynamic multi-constraint graph partitioning." *Concurrency and Computation: Practice and Experience*, 14(3), 2002.

[25] B. R. Stricklanda and D. P. Crowne, "Conformity under Conditions of Simulated Group Pressure as a Function of the Need for Social Approval", *The Journal of Social Psychology*, 58(1), 1962.

[26] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *ACM KDD*, 2010.

[27] A-lias. (2008) Lingpipe 4.0.1. http://alias-i.com/lingpipe.