2017 IEEE 37th International Conference on Distributed Computing Systems

Influence Maximization in a Many Cascades World

Iouliana Litou
Department of Informatics
Athens University of
Economics and Business
Email: litou@aueb.gr

Vana Kalogeraki
Department of Informatics
Athens University of
Economics and Business
Email: vana@aueb.gr

Dimitrios Gunopulos
Department of Informatics
and Telecommunications
University of Athens
Email: dg@di.uoa.gr

***Abstract*—Online Social Networks (OSNs) are widely utilized in viral marketing campaigns exploiting the word-of-mouth effect. Various propagation models have been proposed to describe the way cascades unfold in OSNs. Based on the existing propagation models, several studies address the problem of influence maximization, where the objective is to identify an appropriate subset of users to initiate the spread of a *contagion*. However, existing approaches ignore an important factor in the propagation process, i.e., the *correlation* of multiple contagions simultaneously cascading in the social network and how these affect the users' decisions regarding the adoption of a contagion. Although recent works look into either the competition or the complementarity among a pair of contagions, a uniform model that describes the propagation of multiple cascades with varying types and degrees of correlations is lacking. This work constitutes the first attempt to fill this gap. We formulate a novel propagation model, the *Correlated Contagions Dynamic Linear Threshold (CCDLT)*, that considers the correlation of many contagions in either competitive or complementary manner. Our proposed model allows for different degrees of competition/complementarity among cascades. We further consider that users may dynamically switch states regarding the contagion they promote during the propagation process, based on the influence of their neighborhoods. We then design a greedy *seed selection* algorithm that identifies the appropriate subset of users to participate in a specific contagion in order to maximize its spread and we formally prove that it approximates the best solution at a ratio of $1 - 1/e$. Through an extensive experimental evaluation we demonstrate the superiority of our approach over existing schemes.**

I. Introduction

Social networks play a major role in our daily lives with millions of users exploiting Online Social Services not only as a communication medium but also as a primary channel for news feed. The "word of mouth" effect exhibited in these networks is a very powerful mechanism contributing to the wide adoption of such online services as advertising and influential mediums. With respect to this phenomena, many studies have investigated the problem of influence maximization, where the goal is to identify a small set of influential users to initiate the propagation of a piece of information in the network so that the total number of users reached is maximized [1]–[4]. However, the important factor of *correlated* information being simultaneously diffused in the network, and the impact this presents to the influence spread, is mostly neglected.

Our observation is, that, entities (or information) propagating in the network are not necessarily independent, but present some sort of correlation. For instance, consider the case of political campaigns unfolding in the network of Twitter. Tweets that refer to candidates of competing parties present a negative correlation, as these are conflicting and hence the exposure of a user to both makes the user reluctant in adopting either of them. On the contrary, tweets about the same political parties enhance the propagation of one another, as these are complementary. The same notion of complementarity and competitiveness holds for product advertising as well. For instance, a user that is exposed to the propagation of news regarding both a new version of iOS and a new iPhone model, is keener on searching and probably further disseminating information regarding the new iPhone or the iOS. In that sense, posts of iOS and iPhone are positively correlated and further promote the propagation of one another. On the other hand, consider the case of posts regarding a new Android and a new iOS model. Users may be interested in both, but decide in further propagating information regarding only one of these technologies, or even none, based on the degree of influence by the conflicting information in their network. Therefore, it becomes apparent that propagating entities may be uncorrelated or correlated in a positive/negative manner and that the type of correlation affects the propagation of different entities. The importance of considering the correlation of different contagions in the diffusion process has become evident in recent studies. Myers and Leskovec in [5] performed an analysis in the social network of Twitter[1] to uncover how the exposure of users to different *contagions* (i.e., pieces of information propagating in the network) affects the likelihood of a user adopting a given contagion. Among the most important observations of the study are the following: (i) quantifying how different contagions interact with each other may lead to over 400% more accurate predictions of contagion infections of users in the network, and (ii) contagions cascading in the network present mostly negative interactions, i.e., estimating the propagation of a single contagion while ignoring the presence of other contagions simultaneously cascading in the network may lead to inaccurate estimations of the spread.

Although considerable work has been done in competitive diffusion models in social networks [6]–[11], the correlation of propagating entities is more complex than pure competition. With respect to non-competing items, authors in [12] study the propagation of independent entities. Narayaman et al. in [13] focus on the influence maximization problem of product cross-sell, product-specific costs and benefits and budget constraint, yet their work addresses the special case of solely perfect complementary entities. Lu et al. in [14] were the first to propose a relation between contagions that considers either complementarity or competition under the

[1] https://twitter.com

1063-6927/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDCS.2017.163

Independent Cascade (IC) propagation model. However, their model examines the relation of contagions in pairs of two, which is rather restrictive and insufficient for describing the complexity of many contagions interacting in the network. Finally, McAuley et al. in [15] exploit logistic regression on user reviews to identify substitute/complementary products, however their study performs data analysis and focuses on behavior prediction, without providing an adequate diffusion modeling or addressing the problem of influence maximization.

The lack of models that systematically describe information diffusion of multiple contagions motivated us to design a novel, more expressive and reasonably tractable propagation model to represent the competition and complementarity of propagating entities. We propose therefore the *Correlated Contagions Dynamic Linear Threshold (CCDLT)* model and design an influence maximization approach that considers the correlation of the contagions in the diffusion process. Unlike existing propagation models, CCDLT captures a more complex form of correlation between different contagions, e.g., highly or loosely positively/negatively correlated contagions, and how these affect the propagation of a given contagion. To our knowledge, this is the first work to address the problem of influence maximization while considering two important parameters: (a) the co-existence of *multiple contagions*, and (b) the different *degrees of competitiveness or complementarity* among the contagions. The contributions of this work are summarized as follows:

- We propose and formulate a novel diffusion model, the CCDLT model, that defines and captures the degree of interaction among different contagions propagating in the network. CCDLT is expressive enough to sufficiently describe the way the diffusion of multiple correlated contagions unfolds, allowing different degrees of correlation among the various contagions. Furthermore, the proposed model does not assume that users remain loyal to an adopted contagion, but neither that they randomly switch contagions, achieving thus a more realistic modeling of the propagation process.
- Under the CCDLT propagation model we develop a greedy *seed selection* algorithm that identifies the appropriate subset of users to participate in a specific contagion in order to maximize its spread. Our proposed approach approximates the best solution at a ratio of $1 - 1/e$ and has polynomial worst case complexity $O(m^3)$, where m is the number of users in the network.
- Through an extensive experimental evaluation we illustrate the correctness of the CCDLT propagation model regarding the description of the diffusion process in social networks. We further demonstrate the superiority of our proposed approach for influence maximization compared to existing schemes. Our experimental results show that our approach outperforms its competitors in all cases and independently of the structure of the underlying network, while being orders of magnitude faster.

II. Model and Problem Formulation

In this section we introduce our system model and formulate the problem of influence maximization during the emergence of multiple correlated contagions.

A. System Model

1) Social Graph: The Social Network is represented as a weighted directed graph, referred as *social graph* $G\{N, E\}$. The nodes of the graph N correspond to the users of the social network and the directed edges E between nodes represent the flow of influence, i.e., an edge $e_{uv} \in E$ from user u to v implies that user u influences v. Note that the edge e_{uv} differs from e_{vu}, e.g., in Twitter or Facebook the edge may express the *followee* relationship between users and e_{uv} denotes that user v follows u and hence is influenced by u, but the relation between v and u is not necessarily reciprocal. The weight w_{uv} of the edge e_{uv} denotes the influence strength user u holds over v, i.e., the probability of v adopting contagions that u has previously conveyed. Identifying the appropriate values of w_{uv} is out of the scope of this work, however, existing works such as the approach in [16] which exploits a log of actions to infer social influence, can be applied. Similarly to the work in [2] we assume the sum of incoming edge weights on any node to be at most 1, that is, for every $v \in N$ we have $\sum_{u \in in(v)} w_{uv} \in [0, 1]$. We model the social graph as a $m \times m$ sparse matrix $W_{m,m}$, referred as *social graph matrix*, where m is the total number of users in the network and the value w_{ij} of the $i-th$ row and $j-th$ column denotes the weight of the edge e_{ij}.

2) Contagion Correlation Matrix: We define as *contagion* any information that is spread in the social network regarding a specific topic, e.g., in the network of Twitter or Facebook a contagion may be expressed as the set of posts containing a specific hashtag or URL. For instance, consider Figure 1 that refers to the United States elections and contains the most frequent hashtags paired with the `#debate` hashtag. Each hashtag comprises a separate contagion and the coexistence denotes a positive correlation. However, not all contagions are positively related, e.g., it is unlikely to have the hashtags `#ImWithHer` and `#TrumpWon` together. Moreover, if a user is exposed to both hashtags he/she may hesitate in adopting either of them. It becomes thus clear that contagions may interact with one another during the diffusion. Based on how contagions interact, we define three types of correlation:

Uncorrelated contagions: independent contagions are characterized as uncorrelated. That is, if they both reach a user in the network, the exposure of a user to one of them does not influence the user's perspective on the other.
Positively correlated contagions: two contagions are positively correlated when the exposure of a user to one, positively predisposes the user towards the other contagion, e.g., it is reasonable to assume that posts containing the hashtag `#ImWithHer` are positively related to posts with `#Clinton` as these refer to the same candidate.
Negatively correlated contagions: contrary to the positively correlated contagions, negatively correlated contagions make users reluctant in adopting either contagion when users are exposed to both, e.g., this may be the case for posts supporting different political parties during the election campaigns.

We note that there are different degrees of positive/negative correlations, i.e., `#ImWithHer` and `#TrumpWon` are strongly antagonistic and it can be assumed thus that they are highly negatively correlated, as opposed to hashtags `#Hillary` and

[2]http://hashtagify.me/hashtag/debate

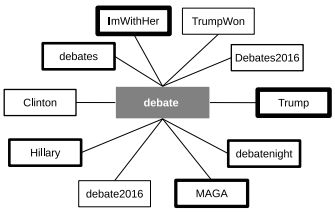


Fig. 1: Correlation of #debate hashtag in Twitter. Hashtags with bolder outline appeared more frequently paired with #debate [2]

#Trump that despite being negatively related, they may be paired in a generic tweet regarding the elections. Hence, using the notation c_i to denote contagion i, we express the degree of correlation between different contagions as a probability and construct the correlation matrix $B_{n,n}$ of n different contagions where $b_{ij} \in B_{n,n}$ denotes the degree to which contagion c_i correlates to contagion c_j. The values of b_{ij} belong to the range of $[-1, 1]$, the absolute value expresses the correlation probability and the signed value the positive or negative correlation. Hence, a value of $b_{ij} < 0$, $b_{ij} > 0$ and $b_{ij} = 0$ denotes that c_i and c_j are respectively negatively, positively correlated or uncorrelated. We assume that the matrix $B_{n,n}$ is symmetric, i.e., $b_{ij} = b_{ji}$. In this work we ignore the sentiment of the context the contagion is presented and therefore any post that the user is exposed to regarding the topic is assumed to be positively related to it, i.e., all posts containing the hashtag #Trump are positively correlated to the contagion of #Trump, even when the users express a negative perspective. Note that estimating the values b_{ij} is out of the scope of this work, but approaches such as the one suggested in [5], where they learn how different contagions interact with each other based on statistical models, may be exploited for this purpose.

In the correlation matrix $B_{n,n}$ we have so far ignored the fact that there are contagions that exhibit high correlation rates. For instance, certain hashtags appear often paired with other entities, e.g., $\#debate$ appears often with candidate related hashtags, as shown in Figure 1. To balance the effect of such generic contagions, we compute the $L1$ norm of $B_{n,n}$, as:

$$||B||_1 = \max_{0 \le j \le n} \sum_{i=1}^{n} |b_{i,j}| \tag{1}$$

and then construct matrix $C_{n,n}$ that is equivalent to $B_{n,n}$, i.e., $B \equiv C$ by dividing all values of B by the corresponding $L1$ norm value $||B||_1$. That is

$$C_{n,n} = \frac{1}{||B||_1} B_{n,n} \tag{2}$$

Based on how $C_{n,n}$ matrix is constructed, it holds that $\sum_{j=0}^{n} |c_{ij}| \le 1$, $\forall i = 0..n$. We later refer to the correlation of contagion i to j as c_{ij}, denoting the value of the $i-th$ row and the $j-th$ column of matrix $C_{n,n}$.

3) User Contagions Matrix: Given the social graph and the different contagions, we denote as *active* those users participating in the propagation of a contagion $c_j, \forall j = 1..n$. Since there are multiple contagions, users may be active in a number of cascades or none. Moreover, during the diffusion process, inactive users can become active or active users may switch contagions and promote a different one. Assume that user u_k is *active* in contagion c_j at time t with probability $u_{kj}(t)$. For all m users and n contagions we can present the activation probabilities of all users over all contagions by constructing a matrix $U_{m,n}(t)$ where $u_{kj} \in U_{m,n}(t)$ denotes the probability of user u_k participating in the propagation of contagion c_j at time t. Thus, matrix $U_{m,n}(t)$ expresses the degree to which users participate in the propagation of the different cascades at time t.

B. Propagation Model

Different propagation models have been proposed to express the way information spreads in social networks. The Independent Cascade (IC) [1], [3] and the Linear Threshold (LT) [1], [2] propagation models are the most prevalent. However, both are limited as they focus on the propagation of a single contagion and assume that whenever users become active they remain loyal throughout the propagation process. Yet, when multiple contagions propagate, users may switch state (*i.e.,* opinion) during the diffusion process regarding the contagions they adopt and further propagate to the network. As existing propagation models fail to capture such phenomena, we introduce a novel propagation model that extends the Linear Threshold model, namely the *Correlated Contagions Dynamic Linear Threshold (CCDLT)* propagation model.

In the CCDLT model, each user u_k is initially associated with a threshold $\theta_{u_k}(0)$ that expresses the user's reluctance in adopting a contagion. At step t, if the incoming influence of a contagion c_j exceeds the users threshold, u_k becomes active at contagion c_j and the threshold increases as:

$$\theta_{u_k}(t) = 1 - (1 - \theta_{u_k}(0))^{y+1} \tag{3}$$

where y is the number of times a user has switched contagions. We assume that in the case that multiple contagions simultaneously exceed the threshold θ_{u_k} of user u_k, the user's threshold is updated, but if there is a negative correlation among the contagions, no contagion is adopted, otherwise all contagions are adopted. The above equation intuitively expresses that whenever a user becomes active over a contagion, then it is less likely to change the adopted contagion. Notice that the more persuadable the user is initially, i.e., low initial threshold, the smaller is the threshold increase. Evidently, low thresholds characterize either early adopters or gullible users. In the case of early adopters the lower rate of increase denotes that they are likely to preserve this tendency. On the other hand, gullible users are more likely to adopt a contagion without particular attention on its validity and hence they are prompt to switching contagions along the process.

A graphic presentation of the system model with the Social Network and many cascades unfolding in the network is depicted in Figure 2. In the example there are three different contagions, $C0, C1$ and $C3$. The correlation matrix of the contagions declares that contagion $C0$ and $C1$ are negatively correlated, while the rest of the contagion pairs present positive interaction, however at a different degree. Green users are active on contagion $C0$, blue on $C1$ and red on $C2$, while users participating in none of the contagions are presented with a gray color. Weights on the edges denote the strength of the influence while the θ values denote the users' thresholds.

C. Problem Definition

Given (i) a Social Graph $G\{V, E\}$, (ii) the correlation of contagions $C_{n,n}$, (iii) the set of active users at time $t = 0$ over the different contagions $U_{m,n}(0)$, (iv) a budget x and

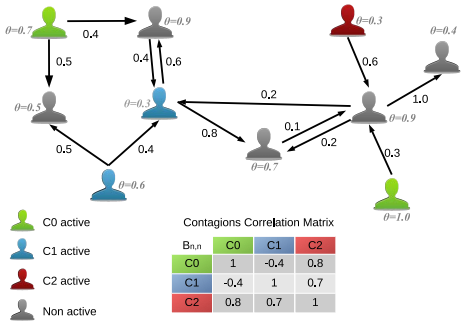


Fig. 2: Example of a Social Graph along with the Correlated Contagions and participants of each contagion

(v) a contagion c_j, our goal is to identify a subset of users $S = \{u_1, .., u_x\}$ of size x, referred as seeds, such that the propagation of contagion c_j under the CCDLT model is maximized when $u_{ij}(0) = 1, \forall u_i \in S$. During the diffusion we assume that seeds may not switch between contagions. We refer to this problem as *influence maximization with multiple correlated contagions*. The above problem can be proven to be NP-Complete, as influence maximization given the simple LT propagation model is a special case of the CCDLT where $C_{n,n}$ contains a single entry.

III. Correlated Contagions Dynamic Linear Threshold Model- CCDLT

In this section we present a thorough analysis of our influence estimation model. We first show how the influence of correlated contagions is estimated for a single user and then how to apply these computations in the entire network. Afterwards, we provide a detailed description of our seed selection approach for influence maximization.

A. Influence Probability Estimation

In the network, the influence of a contagion c_j to a user u_k is derived from the incoming neighbors of u_k. The strength of the influence of c_j depends on both the influence of each of the in-neighbors of u_k that are active at c_j, as well as the correlation of c_j to the different contagions that the neighborhood propagates.

1) Influence Estimation of a Single Contagion: For now let us assume that there is no correlation among contagions, and that users are either active or inactive during the cascade. Hence, given $U_{m,n}(t-1)$ we assume that each user u_k has at most one $u_{kj}(t-1) = 1, \forall j = 1..n$, otherwise $u_{kj}(t-1) = 0$. Given the activation probabilities matrix $U_{m,n}(t-1)$ at time $t-1$, the social graph matrix $W_{m,m}$ and a contagion c_j, the influence probability $f_k(c_j|t)$ of user u_k on contagion c_j at step t can be computed as

$$f_k(c_j|t) = \sum_{u_i \in in(u_k)} w_{ik} \cdot u_{ij}(t-1) \quad (4)$$

where $in(u_k)$ is the set of neighbors with incoming edges to u_k, w_{ik} is the weight of the edge e_{ik} and $u_{ij}(t-1)$ is the probability of user u_i participating in contagion c_j at time $t-1$. Intuitively, when ignoring correlations among the contagions, the influence probability of contagion c_j on user u_k at step t depends on the user's incoming influence from neighbors that are active at contagion c_j at a prior step. In essence, the users in social networks often share content that their friends have previously published, hence the influence from the neighborhood expresses the probability of a user adopting a post published by the immediate incoming neighborhood. As there may be multiple concurrent contagions cascading in the network, for each user u_k, at time t we maintain a vector $\vec{u_k}(t) = \begin{pmatrix} f_k(c_1|t) & \cdots & f_k(c_n|t) \end{pmatrix}$, where $f_k(c_j|t)$ denotes the influence of c_j on u_k from the user's neighborhood.

2) Influence Probability for Correlated Contagions: Equation 4 assumes independence of concurrent contagions. However, depending on the correlation of the various contagions, the exposure of the user to contagion c_l may boost or decrease the influence of c_j. Therefore, to estimate the final influence probability $p(u_i|c_j)(t)$ of the contagion c_j we also consider the correlation of $c_l, \forall l = 1..m$ to c_j, by multiplying the vector $\vec{C_j}^T$, that represents how c_j is correlated to other contagions, with vector $\vec{u_k}(t) = (f_k(p_1|t) \quad \cdots \quad f_k(p_n|t))$, that denotes the social influence exerted to the user over all contagions. Therefore, the influence probability $p(u_i|c_j)$ is expressed as

$$p(u_k|c_j) = \vec{C_j}^T \vec{u_k}(t) = \sum_{l=1}^{n} c_{l,j} \cdot f_k(c_l|t) \quad (5)$$

Thus, the influence probability $p(u_k|c_j)$ of contagion c_j to user u_k is computed as the sum of the correlations of c_j to any other contagion c_l while considering the influence probability $f_k(c_l|t)$ of c_l to u_l, as this is derived from the user's neighborhood (Equation 4). Note that $p(u_k|c_j)$ may take negative values. If $p(u_k|c_j)$ is negative, this indicates that the user is negatively influenced by c_j and hence does not adopt the propagation. However, if the absolute value of $p(u_k|c_j)$ exceeds the user's threshold $\theta_{u_k}(t)$, then the user may act as a blocker in the spread of the contagion c_j. Therefore, the user may also have a negative impact on the propagation of a contagion. If the user is negatively predisposed to a specific contagion, i.e., $p(u_k|c_j) < 0$, she is undermining instead of promoting the spread of the contagion c_j and presents a positive impact on the propagation of negatively correlated contagions. Due to the multiplication with the correlated contagions vector the above sum may return non-zero $p(u_k|c_j)$ values for contagions that the user has not been exposed to within the network. We exclude such cases by setting a zero $p(u_k|c_j)$ value to the contagions that none of the incoming neighbors of a user participates to.

3) Influence Estimation in the Entire Network: In the previous sections we described how the influence of a single contagion is estimated for a single user. In this section we explain how the influence probability at each step t may be estimated in the entire network and for all contagions. Given the social graph matrix $W_{m,m}$ and the activation probabilities matrix $U_{m,n}(t-1)$ at step $t-1$, the influence $f_k(c_j|t) \forall k = 1..m, j = 1..n$ can be estimated as the product of the matrices $W^T_{m,m}(t)$ and $U_{m,n}(t)$. That is,

$$F_{m,n}(t) = W^T_{m,m} \cdot U_{m,n}(t-1) \quad (6)$$

Equation 6 returns the influence probabilities of all contagions in the network when correlations are ignored, i.e., table $F_{m,n}(t)$ contains the vectors $\vec{u_k}$ for all users $u_k, k = 1..m$. To consider the correlation of the different contagions we multiply $F_{m,n}$ with the contagions correlation matrix $C_{n,n}$ (since the $C_{n,n}$ is symmetric, $C_{n,n} = C^T_{n,n}$). Hence, the final influence probability of all contagions in the network when correlations are considered, is estimated as

$$A_{m,n}(t) = W^T_{m,m} \cdot U_{m,n}(t) \cdot C_{n,n} \quad (7)$$

B. Influence Maximization Algorithm

In this section we describe our algorithm to addressing the problem of influence maximization of a target contagion c^* under the CCDLT propagation model. Since the influence maximization problem with multiple contagions is NP-complete the optimal seed set cannot be found in polynomial time. Therefore we design a greedy algorithm that approximates the optimal solution at a ratio of $1 - 1/e$.

Algorithm 1 summarizes the greedy seed selection approach. The overall seed selection process comprises of two main phases, the CCDLT influence computation (lines 6-11) and the CCDLT seed selection given the estimated influence values (lines 12-18), that iteratively selects nodes to be added as seeds. Initially the seed set S is empty (line 1) and negatively correlated contagions $c_l, \forall l = 1..n$ to c^* are identified (line 2). We consider that users in the network participating in competing contagions or users already participating in the propagation of c^* may not be selected as seeds (line 7). In order to avoid an exhaustive estimation of the influence for all nodes in the graph, we first compute the vertex cover V in the undirected graph, similarly to [2]. For nodes $u_k \in V$ we then estimate the expected spread $\alpha(u_k)$ based on the CCDLT propagation model (line 10) and the influence of u_k on contagion c^* (line 11). At each iteration, the node u_i with the maximum estimated influence is selected (line 13). Given the expected spread estimation $\alpha(u_i)$ of node u_i, nodes I that are estimated to participate in the cascade of contagion c^* when u_i initiates the propagation of c^* (estimated by the CCDLT propagation in line 10) are not considered as candidate seeds. The elimination is achieved by setting their influence equal to 0 (line 17). We choose to do this as it is expected that these nodes will eventually be participating in the cascade of c^*.

Algorithm 1: CCDLT Seeds Selection

```
   Data: W_{m,n}, C_{n,n}, U_{m,n}(0), x, c*
   Result: S
1  S → ∅ ;
2  N → negativeCorrelatedCascades(c*) ;
3  V → VertexCover(W_{m,n});
4  α → ∅;
5  influence → ∅;
6  for u_k ∈ V do
7      if u_k(0) ∩ N ≠ ∅ ∨ u_k(0).c* > 0 then
8          continue;
9      if u_i ∉ α then
10         α(u_k) → ccdlt(u_k, η, c*, U_{m,n}(0), C_{n,n}) ;
11     influence(u_k) → InfluenceEstimation(W_{m,n}, u_k, V, α, c*);
12 for i = 1..x do
13     u_i → {u_i : influence(u_i) > 0 ∧
         influence(u_i) ≥ influence(u_j) ∀j = 1..m} ;
14     S → S ∪ u_i;
15     I → influencedBy(u_i, α(u_i)) ;
16     for v ∈ I do
17         influence(v) → 0 ;
18     influence(u_i) → 0;
```

1) Expected Spread Estimation : Algorithm 2 presents the estimation of the expected spread for the first part of the algorithm, when node u_k acts as a seed of the targeted cascade c^*. To estimate the influence of u_k we focus on the neighborhood of u_k, since according to the findings in [17], under the LT model, the expected influence spread of a node becomes negligible after a small number of hops in many real-world social networks. The result is also valid for the CCDLT propagation model as this constitutes an extension of the LT propagation model. Therefore, we construct the subgraph $g(u_k) \subseteq G(N, E)$ that contains all immediate out-neighbors of u_k and their incoming neighbors (line 2). Afterwards, the CCDLT propagation unfolds on subgraph $g(u_k)$. To compute the expected spread of u_k given c^*, we initially set the value $u_{kc*} = 1$ for $u_{kc*} \in U_{m,n}(0)$, i.e., we set that the user u_k is activated over $c*$. For nodes $m' \in g(u_k)$ we estimate the expected spread according to Equation 7 (line 5). Notice that we do not need the entire matrix of users contagions $U_{m,n}$, but the sub-matrix $U'_{m',n}$ containing the nodes m' in the subgraph $g(u_k)$. We first normalize the values of matrix $U'_{m',n}(t-1)$ that represent the activation probabilities so that $\sum_{j=1}^{n} p(u_i|c_j) = 1$, $\forall u_i \in m'$ (line 4). Intuitively, the normalization denotes that if a user is active in more that one contagions (non-competing), then there is a probability of influencing his/her neighbors in all contagions, however, the probability is not necessarily the same for all. In the Social Networks this may be interpreted based on the way posts are presented to the users' timelines, e.g., the most recent posts are more likely to be seen by the followers and further propagate into the network. For users that have not been exposed to a specific contagion c at step t, but present values different than those at $t-1$ for contagion c, we restore the values of $t-1$, as we assume that a user can only be influenced if she is exposed to the contagion. Additionally, we also restore the values if no contagion exceeds the users' threshold or in the case that competing contagions exceed the users' threshold simultaneously (line 6), as in these cases the users are not influenced. Since we only examine the immediate neighborhood of a user, i.e., one propagation step, we do not update thresholds during the seed selection.

Algorithm 2: ccdlt

```
   Data: u_k, c*, U_{m,n}(t − 1), C_{n,n}
   Result: α(u_k)
1  α(u_k) → ∅;
2  g(u_k) → subgraph(u_k, η) ;
3  u_{k,c*}(t − 1) = 1;
4  U_{m,n}(t) → normalize(U_{m,n}(t − 1));
5  A(t) → g(u_k)^T · U'_{m',n}(t − 1) · C_{n,n} (Equation 7) ;
6  U'_{m',n}(t) → restoreNonExposed(U'_{m',n}(t − 1), A(t));
7  α(u_k) → normalize(U'_{m',n}(t));
```

2) Influence Estimation: Algorithm 3 summarizes the final influence value computation of a candidate seed, given the expected spread estimation $a(u_k)$. The value is estimated as

$$a(u_k).value = \sum_{i=0}^{m} p(u_i|c^*) + |I| \tag{8}$$

where $p(u_i|c^*)$ is the estimated influence probability of contagion c^* at user u_i in the $a(u_k)$ and $|I|$ is the absolute number of users u_i for whom it holds $p(u_i|c^*) \geq \theta_{u_i}(t)$ and $p(u_i|c^*) \geq p(u_i|c_j)$, $\forall c_j, j = 1..n \neq c^*$. Notice that the set I is the same as that in Algorithm 1 (line 15). Hence, the influence is defined as the sum of the resulting influence probabilities for all nodes, as these are estimated through the $ccdlt$ subroutine, plus the absolute number of users whose influence probability on contagion c^* is greater than any other contagion they may be participating. Intuitively, in the estimation of influence of u_k we consider not only the probability

of a node u_i being influenced by c^* when u_k is selected as a seed, but also the fact of c^* being the prevalent contagion for node u_i. After the influence value of u_k is estimated, for each of the immediate neighbors u_i of u_k we compute the expected spread of u_i if this is not previously estimated (lines 4 - 5) and add the value of the expected spread of u_i to u_k, by multiplying it with the probability of u_k influencing u_i, expressed by the weight of the edge w_{ki} between u_k and u_i.

Algorithm 3: Influence Estimation

Data: $W_{m,n}$, u_k, V, α, c^*
Result: $influence$
1 $influence(u_k) \rightarrow a(u_k).value$;
2 $outNeighbors \rightarrow getOutNeighbors(W_{m.n}, u_k)$;
3 **for** $u_i \in outNeighbors$ **do**
4 **if** $u_i \notin \alpha$ **then**
5 $a(u_i) \rightarrow ccdlt(u_i, \eta, c^*, U_{m,n}(0), C_{n,n})$;
6 $influence(u_k) \rightarrow influence(u_k) + w_{ki} \cdot a(u_i).value$;

C. Complexity Analysis

The worst case complexity arises in the case of a complete graph, as Algorithm 1 computes the expected influence spread for all nodes, however, only once for each user. The complexity of the expected spread estimation equals the complexity of the multiplication of the subgraph matrix $g(u_k)$ with the users' adopted contagions $U_{m,n}(t)$ and the contagion correlation $C_{n,n}$ matrices. In the case of a complete graph $g(u_k) = W_{m,m}$, therefore line 5 of Algorithm 2 has a complexity $O(m^2n)$ to compute $F_{m,n}(t)$ and $O(mn^2)$ to compute the final matrix $A(t)$, resulting in an overall complexity of $O(m^2n + mn^2)$ for the multiplication. Restoring the values of non exposed users and the normalization presents a complexity of $O(m)$ and $O(mn)$ respectively. The multiplication is repeated for all m users by setting a different $U_{m,n}(t)$. Therefore the overall complexity equals $O(m(m^2n + mn^2 + mn + m)) \approx O(m^3n + m^2n^2)$. Usually the number of users to the network m is orders of magnitude greater than the number of the contagions n, i.e., $m \gg n$, and hence it suffices to express the overall complexity as $O(m^3)$. The complexity can be reduced if we limit computations in smaller graphs of strongly connected communities [18].

D. Approximation Ratio

In Appendices A and B we prove the monotonicity and submodularity properties respectively and thus that the seed selection has an $1 - 1/e$ approximation ratio.

IV. Experimental Evaluation

In this section we illustrate the necessity of considering interactions of contagions in the process of identifying influential users. We demonstrate how existing approaches overestimate the influence of a user when the correlation among contagions is ignored and showcase the efficiency of our approach under (i) various types of correlations, i.e., complementary/competitive contagions, (ii) various sizes of users participants in the cascades, and (iii) various number of correlated contagions. Moreover, we demonstrate the superiority of our methodology for seed selection in terms of execution times and memory consumption. For the evaluation we exploit

	Twitter	HEP-PH
Nodes	23988	12007
Edges	20385	237008
Average Degree	1.7	39.478
Network Diameter	10	13
Average Path Length	2.232	4.67
Modularity	0.97	0.656

TABLE I: Graph characteristics

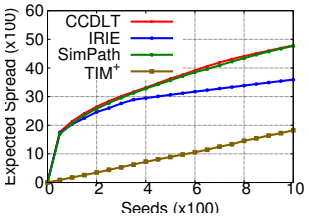


Fig. 3: Expected spread of a single cascade in the network of Twitter

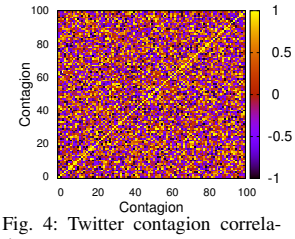


Fig. 4: Twitter contagion correlations

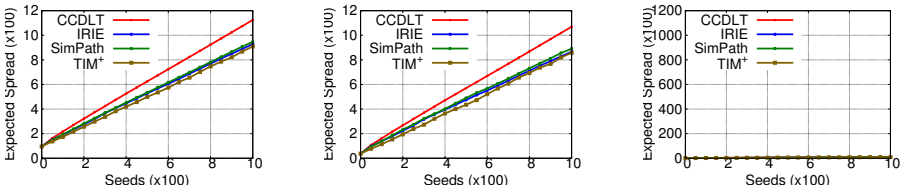
two real-world networks, a network derived from *Twitter* and the *High-Energy Physics (HEP-PH)* citation network [19]. We compared our seed selection approach to three well-known state-of-the art algorithms:

SimPath [2]: SimPath is a greedy seed selection algorithm for the problem of Influence Maximization under the Linear Threshold propagation model. The spread estimation of a node in SimPath is computed by summing the weights of all simple paths originating from it. Optimizations are used to avoid the exhaustive spread computations for all nodes in the graph, i.e., the vertex cover is utilized to estimate the influence and a look-ahead optimization is further exploited.
IRIE [3]: IRIE is an iterative greedy algorithm that uses Influence Ranking and Influence Estimation. At each iteration the node with the largest marginal spread, as computed through Monte-Carlo simulations, is added to the seed set. Once a seed is added its additional influence is estimated.
TIM⁺ [4]: TIM is Two-phase Influence Maximization algorithm. In the first phase the lower-bound of the maximum expected spread among all size-k node sets is estimated and it is later exploited in the second phase of the node selection. An optimization of it is TIM⁺, where an intermediate step is introduced in order to acquire a tighter lower-bound estimation. We set the values ϵ and l required in the algorithm as these are suggested in the corresponding study, i.e., $\epsilon = 0.2$ and $l = 1$.

Experimental Setup. We assign user thresholds uniformly at random, following the work of [20]. The influence probabilities among the users, i.e., the weights of the edges, are assigned according to [4]. That is, for each incoming node u to v, we initially generate a random probability in $[0, 1]$ for the edge e_{uv} and then normalize the incoming probabilities to v so that they sum up to 1. With reference to the correlation of contagions, unless otherwise specified, we assign correlations randomly but with a slightly negative probability, as the work in [5] discovered that there appears to be a default negative interaction between contagion pairs. Specifically, we set the ratio of negatively correlated contagions to be 15% higher than the positively correlated contagions. Finally, contagions are randomly assigned to users to construct the $U_{m,n}(0)$ matrices.

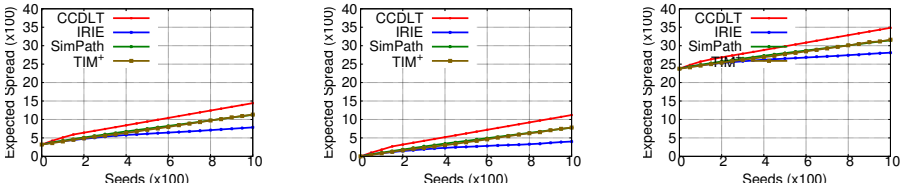
A. The Twitter Network

The first network we investigate is derived from the social network of *Twitter*. We used a set of tweets collected from



(a) Average (b) Minimum (c) Maximum

Fig. 5: The average, minimum and maximum estimated spread in Twitter from all 100 cascades and 40% of the users participating in one



(a) Average (b) Minimum (c) Maximum

Fig. 6: The average, minimum and maximum estimated spread in Twitter from all 100 cascades and 100% of the users participating in one

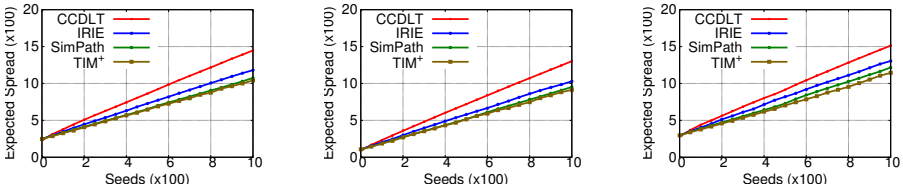
the area of UK in January 2014 [21], where we identified the top 100 most popular hashtags and users that published tweets containing these hashtags. Based on the mentions and replies to the tweets of the filtered users we constructed the network of users. Filtering out the users based on the content they publish to construct the social graph is a methodology also used in [5], but unlike their approach that exploits the URLs, we rather consider hashtags. The final network comprises of approximately 24K users and 20K edges. Information regarding the graph characteristics is presented in Table I. A graphic representation of the randomly generated correlation of contagions is depicted in Figure 4, with darker hues implying negative correlations, while the lighter ones denote positive correlations. The degree of the correlation is denoted by the intensity of the hue. The figure is symmetric to the diagonal that denotes the correlation of each contagion to itself.

In the first set of the experiments we evaluated the different seed selection approaches for the traditional problem of the influence maximization, where there is a single contagion whose spread we aim to maximize in the network and no users are initially active. We present the results in Figure 3, where we observe that our approach and SimPath perform equally well and are better compared to the rest of the approaches. TIM$^+$ performs overall poorly, which may be due to the Reverse Reachable (RR) sets produced. As Twitter is a sparse graph, there may not be sufficient covers between the different RR sets. As the results indicate, our approach effectively addresses the traditional Influence Maximization problem, yet the most interesting aspect of the approach is revealed when the correlation among the contagions is considered and the number of participants in the contagions varies, as we show next. We first show the results with 40% of the users participating in the cascades before seeds are selected (Figure 5), and then with all 100% of the users being active in one of the 100 contagions (Figure 6). We deploy the seed selection algorithms for all contagions. The average, maximum and minimum expected spread from all contagions are shown in the figures. Since there are already users in the network active at different cascades, the expected spread when seeds are absent is greater that 0. Notice that compared to Figure 3, the expected spread drops dramatically when there are correlations among contagions, which is in accordance to the findings of [5], suggesting that the proposed propagation model effectively captures the propagation trends of multiple correlated contagions. Moreover, the CCDLT seed selection algorithm for influence maximization outperforms its competitors in all cases. When the portion of initially active users increases from 40% to 100% (Figure 5), so does the overall expected spread for the contagions, as more users participate in each cascade. Still, the expected spread for seeds selected by the CCDLT algorithm is greater by up to 16% compared to the rest of the approaches. SimPath and TIM$^+$ follow and perform equally well, while IRIE falls behind as the number of users engaged into the cascades increases.

B. High-Energy Physics Collaboration Network - HEP-PH

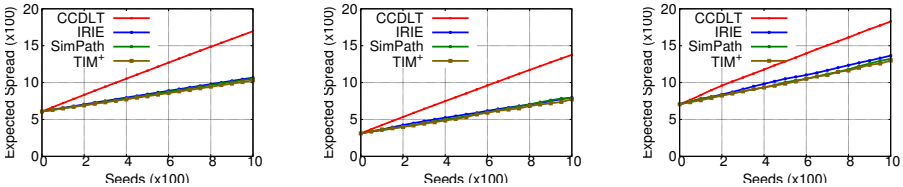
The second network we use is derived from the High-Energy Physics (HEP-PH) collaboration network [19]. HEP-PH is a collaboration network that is often exploited in studies regarding Influence Maximization [4], [22]. The nodes of the graph represent authors and the edges denote co-authorships. The network initially contains undirected edges, therefore for each edge between nodes u and v, we construct the graph by adding directed edges e_{uv} from node u to v and e_{vu} from node v to u. The final graph consists of 12K nodes and 237K edges. Information about the graph structure is presented in Table I. For the network of HEP-PH we conducted experiments with (i) a single contagion and no users initially active (Figure 9), (ii) 20 randomly correlated contagions with 40% (Figure 7) and 100% (Figure 8) of the users in the network participating in the cascade of contagions, (iii) 20 complementary and 20 competitive contagions with 40% of the users participating in the propagation (Figures 11 and 12 respectively).

In Figure 9 we notice, that, contrary to Twitter, in the HEP-PH network the IRIE algorithm performs better than the rest with CCDLT being close by and SimPath following, while TIM$^+$ still falls behind. In this case, the poor performance of TIM$^+$ is caused by the exclusion of RR sets, leaving seeds having few paths with low probabilities to influence the rest of the nodes in the network. Still, we note that the above results correspond to the case of a single contagion propagating in the network, which leads to overestimation of the expected influence probability, as the correlations with other contagions in the network are ignored. Having a single



(a) Average (b) Minimum (c) Maximum

Fig. 7: The average, minimum and maximum estimated spread in HEP-PH from all 20 cascades with 40% participating



(a) Average (b) Minimum (c) Maximum

Fig. 8: The average, minimum and maximum estimated spread in HEP-PH from all 20 cascades with 100% participating

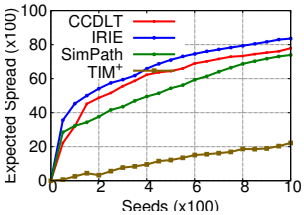


Fig. 9: Expected spread of a single cascade in the network of HEP-PH

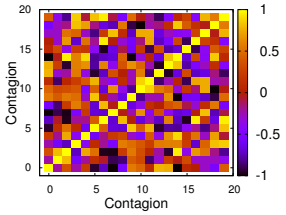


Fig. 10: HEP-PH 20 cascades correlation

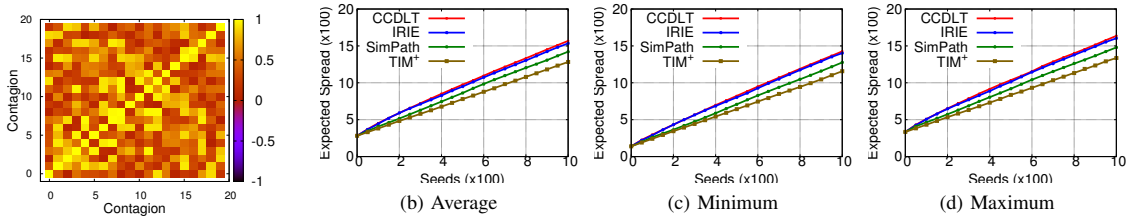
contagion propagating in a network of billions of users is quite an unlikely scenario. Notice that when correlated cascades are present the performance of IRIE drops, while CCDLT outperforms the rest of the competitors by up to 19% as depicted in Figures 7 and 8. The correlations of the cascades are presented in Figure 10. Similarly to the Twitter network, the overall propagation performance drops for all of the approaches, with the gap between CCDLT and the competitive approaches increasing as the number of participants in the contagions rises. In the case of complementary contagions, i.e., $b_{i,j} \geq 0$, $\forall b_i$, b_j pairs of contagions (Figure 11a), we notice that all approaches perform slightly better, with IRIE performing equally well to CCDLT (Figure 11), yet the expected spread is lower compared to the case of a single cascade, confirming the default negative interaction between contagion pairs identified in [5]. Examining this behavior under the mathematical aspect of our proposed propagation model, when contagions are positively related users may participate in the propagation of more than a single contagion. However, as many contagions are adopted, the degree to which the users support the propagation of a specific contagion drops (captured by the normalization of the $U_{m,n}(t)$ matrix) and hence the overall influence of the contagion decreases. The results for the competitive contagions, i.e., $b_{i,j} \leq 0$, $\forall b_i$, b_j (Figure 12) prove that CCDLT performs better than existing approaches with almost constant performance compared to Figure 7.

C. Overhead Evaluation

We further conducted a set of experiments to show how the different approaches perform in terms of execution times and memory usage. Our experiments are conducted on a machine with an Intel Core i7-3770 CPU at 3.4GHz, running 64bit Ubuntu 14.04. All algorithms are implemented in Java 8. We demonstrate the results for the case of the HEP-PH network with 20 randomly correlated contagions and 40% of the users being initially active on one of them. The execution times are presented in Figure 13. Our results suggest that for a small number of seeds, i.e., fewer than 50 seeds, IRIE is faster than the rest of the approaches. However, as the number of seeds increases the execution time of IRIE increases exponentially. SimPath also presents an exponential increase in execution times as the number of seeds increases, however the ratio of increase in lower than that of IRIE. On the contrary, the CCDLT and TIM$^+$ seed selection process require a constant time for the initial estimations and thereafter the seed selection algorithm requires almost constant time. CCDLT is yet faster, requiring less than 7 minutes to identify the appropriate seed set. Therefore, we observe that CCDLT not only selects a better seed set when correlated contagions emerge, but does so while running orders of magnitude faster. With reference to the memory usage, we notice in Figure 14 that CCDLT has a constant memory usage, TIM$^+$ memory usage increases in proportion to the seed size, while IRIE and Simpath present variations. Specifically, IRIE presents high memory consumption followed by sudden drops. These indicate calls to the Java Garbage Collector. The fact the CCDLT presents a constant memory consumption denotes that our approach requires significantly less memory compared to the rest at any time, therefore no calls are required to free memory space. Since memory usage variations depend on the implementation, we note that we have implemented the competing approaches strictly following the description presented in the corresponding literature.

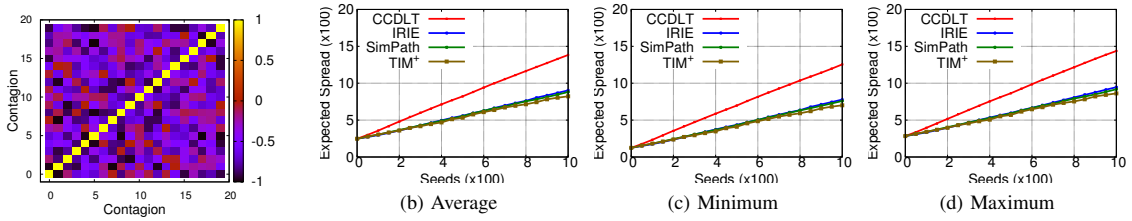
V. Related Work

Authors in [23] model the propagation of multiple competing products. They define three types of product adoption by users, namely the self and social adoption and the social conversion. Yet their model only captures competition and hence fails to appropriately describe the propagation in social networks, where co-operation of contagions is also present. In [24] authors expand the Independent Cascade model in order to capture competitiveness of propagating entities in



(a) Correlations (b) Average (c) Minimum (d) Maximum

Fig. 11: The average, minimum and maximum estimated spread in HEP-PH from all 20 **complementary** cascades with 40% participating



(a) Correlations (b) Average (c) Minimum (d) Maximum

Fig. 12: The average, minimum and maximum estimated spread in HEP-PH from all 20 **competitive** cascades with 40% participating

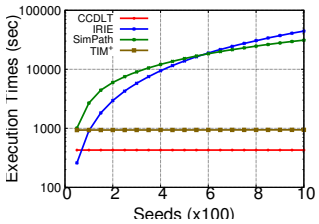


Fig. 13: Execution Times in HEP-PH with 20 contagions and 40% participants

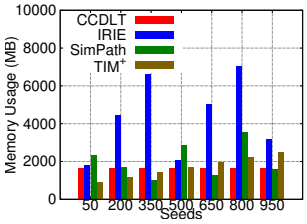


Fig. 14: Memory Usage in HEP-PH with 20 contagions and 40% participants

the diffusion process. However their approach models how users may interact with the different contagions, with contagions being independently adopted and thus failing to capture correlation of different contagions. Authors in [25] study the information spreading mechanism of two type of messages over networks, the competitive and the cooperative types. Yet, in their work the problem of seed selection to enhance the propagation of a specific contagion is not addressed. Borotin et al. in their work [7] propose extensions of the Linear Threshold propagation model to formulate competitive influence cascades in the network. Authors in [26] study the problem of seed allocation from the perspective of a host, in order to guarantee the fairness of competition to antagonistic customers exploiting the network for viral campaigns. Authors in [27] suggest a methodology to identify an appropriate subset of nodes in order to maximize the influence probability to a target node. However, in their study they ignore correlated contagions users along the paths are exposed to, which leads to inaccuracy in the estimations of the appropriate routes, as shown in [5]. Authors in [14] propose a complementarity relation between pairs of contagions under the Independent Cascade (IC) propagation model. In [5], authors examine how different contagions interact with each other as they spread through the network. A statistical model is proposed that allows for competition as well as cooperation of different contagions in information diffusion. A mixing model is developed, where the probability of a user adopting a piece of content is based on what other content the users was previously exposed. It is assumed that the infection probability is similar for all users. He and Kempe in [28] study the problem of robust influence maximization under uncertainty of the propagation model and aim at identifying the appropriate seeds that are simultaneously influential for all influence functions. Chen *et al.* in [29] propose a seed selection approach that considers the uncertainty of the influence probability estimations in the edges. In [30] authors investigate how item inference relates to social influence and suggest a seed selection methodology in order to maximize the total adoptions of a set of products. Authors in [31] introduce a mathematical model that assists in determining the influence of users. Ohsaka *et al.* in [32] propose a dynamic index data structure for influence analysis on evolving networks and present algorithms for queries on influence estimation and influence maximization. Still, correlations of contagions and how these impact on the influence estimations is ignored in the majority of the literature. In [33] the authors predict the item-level social influence that can be exploited to identify influential users regarding a specific contagion.

VI. Conclusions and Future Work

We formulated a novel propagation model, the Correlated Contagions Dynamic Linear Threshold (CCDLT), that describes how multiple correlated contagions propagate in the network. This is the first attempt to formulate the correlation of many cascades in Social Networks. The results of the study suggest that CCDLT effectively captures the way diffusing contagions interact. We designed an influence maximization algorithm to select the appropriate seed set to maximize the spread of a given contagion and proved its superiority to existing approaches. There are interesting directions to further research, e.g., (i) multiple strategic players concurrently aiming to maximize their spread, and (ii) influence maximization in a many cascades world with evolving networks.

Acknowledgement

This research has been financed by the European Union through the FP7 ERC IDEAS 308019 NGHCS project and the Horizon 2020 688380 VaVeL project.

APPENDIX A
PROOF OF MONOTONICITY

The influence of contagion c_j to the immediate neighborhood of a node u_k is estimated as

$$\begin{aligned} I(c_j|u_k)(t) &= \sum_{u_i \in N^{out}(u_k)} \sum_{l=1}^{n} c_{lj} \sum_{u_{i'} \in N^{in}(u_i)} w_{i'i} \cdot u_{i'l}(t-1) \\ &= \sum_{u_i \in N^{out}(u_k)} \sum_{l=1}^{n} c_{lj} \sum_{u_{i'} \in N^{in}(u_i) \setminus u_k} w_{i'i} \cdot u_{i'l}(t-1) \\ &+ \sum_{u_i \in N^{out}(u_k)} w_{ki} \Big(\sum_{l \neq j} c_{lj} u_{kl}(t-1) + c_{jj} u_{kj}(t-1) \Big) \end{aligned}$$

The additive value $\sigma_j(u_k)$ of node u_k participating in the contagion c_j that u_k was not previously propagating, i.e., $u_{kj}(t-1) = 0$, is expressed in the second part of the last equation, as $\sum_{u_i \in N^{out}(u_k)} w_{ki}$ remains constant.

$$\sigma_j(u_k) = \sum_{u_i \in N^{out}(u_k)} w_{ki} \Big(\sum_{l \neq j} c_{lj} u_{kl}(t-1) + c_{jj} u_{kj}(t-1) \Big) \quad (9)$$

When u_k does not participate in c_j it holds $c_{jj} u_{kj}(t-1) = 0$. After selecting u_k as a seed, $u_{kj}(t-1) = 1$. The normalization afterwards results in values $u'_{kl}(t-1) \leq u_{kl}(t-1)$, implying that $\sum_{l \neq j} c_{lj} u'_{kl}(t-1) \leq \sum_{l \neq j} c_{lj} u_{kl}(t-1), \forall l = 1..n$. Therefore, we have to prove that

$$\sum_{l \neq j} c_{lj} u_{kl}(t-1) \leq \sum_{l \neq j} c_{lj} u'_{kl}(t-1) + c_{jj} u'_{kj}(t-1) \quad (10)$$

Proof: Since initially $u_{kj}(t-1) = 0$, it applies that

$$\sum_{l=1}^{n} u_{kl}(t-1) = \sum_{l \neq j} u_{kl}(t-1) = 1 \quad (11)$$

In order to acquire the vector $u'_k(t-1)$ we divide the elements of the resulting vector after setting $u_{kj}(t-1) = 1$ to their sum, i.e.,

$$u'_{kl}(t-1) = \frac{u_{kl}(t-1)}{u_{kj}(t-1) + \sum_{l \neq j} u_{kl}(t-1)} = \frac{u_{kl}(t-1)}{2} \quad \forall l = 1..n \quad (12)$$

Equation 12 implies that $u'_{kj}(t-1) = \sum_{l \neq j} u'_{kl}(t-1) = \frac{1}{2}$. Applying Equation 12 to Equation 10 results in

$$\begin{aligned} \sum_{l \neq j} c_{lj} u_{kl}(t-1) &\leq \sum_{l \neq j} c_{lj} \frac{u_{kl}(t-1)}{2} + \frac{1}{2} c_{jj} \\ \Rightarrow \sum_{l \neq j} \frac{c_{lj}}{c_{jj}} u_{kl}(t-1) &\leq 1 \end{aligned}$$

Based on the way the matrix $C_{n,n}$ is constructed it applies that $|c_{jj}| \geq |c_{lj}|$, $\forall l = 1..n$ and $\sum_{j=0}^{n} |c_{ij}| \leq 1$, thus $\frac{|c_{lj}|}{|c_{jj}|} \leq 1$. Additionally, we only consider seeds u_k that present no negatively correlated contagions to c_j, hence, in case $c_{lj} < 0$, $u_{kl}(t-1) = 0$. Therefore $\sum_{l \neq j} \frac{c_{lj}}{c_{jj}} u_{kl}(t-1) \geq 0$. Finally, since $\sum_{l \neq j} u_{kl}(t-1) = 1$ we prove that the inequality holds. ∎

APPENDIX B
PROOF OF SUBMODULARITY

In order to prove the submodularity property, for two sets T and S with $T \subseteq S$ and a candidate seed $u_{k'}$ we have to prove that

$$\sigma_j(S \cup u_{k'}) - \sigma_j(S) \leq \sigma_j(T \cup u_{k'}) - \sigma_j(T) \quad (13)$$

Proof: For nodes u_k and $u_{k'}$ in S, it may hold that $out(u_k) \cap out(u_{k'}) \neq \emptyset$. Let $out(S)$ be the set of all nodes reached by S. The value of S to the cascade of c_j can be estimated as

$$\sigma_j(S) = \sum_{u_i \in out(S)} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \quad (14)$$

Equation 14 estimates the value of each $u_k \in S$, however the value of a node $u_i \in out(u_k)$ is estimated only once to the overall value $\sigma_j(S)$. The value of a candidate seed $u_{k'}$ to be added to S can be then estimated as

$$\sigma_j(S \cup u_{k'}) = \sum_{u_i \in out(S \cup u_{k'})} \sum_{u_k \in (S \cup u_{k'})} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \quad (15)$$

The set on nodes $out(S \cup u_{k'})$ equals $out(S) \cup (out(u_{k'}) \setminus (out(S) \cap out(u_{k'})))$. Exploiting this to Equation 15

$$\begin{aligned} \sigma_j(S \cup u_{k'}) &= \sum_{u_i \in out(S)} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &+ \sum_{u_i \in out(S)} w_{k'i} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) + \sum_{\substack{u_i \in \\ out(u_{k'})}} w_{k'i} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &+ \sum_{u_i \in out(u_{k'})} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &- \sum_{u_i \in (out(S) \cap out(u_{k'}))} \sum_{u_k \in (S \cup u_{k'})} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \end{aligned} \quad (16)$$

For nodes $u_i \in out(S)$ with no incoming edge from $u_{k'}$, it holds $w_{k'i} = 0$

$$\sum_{u_i \in out(S)} w_{k'i} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) = \sum_{\substack{u_i \in (out(S) \\ \cap out(u_{k'}))}} w_{k'i} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \quad (17)$$

Similarly, if a node u_i in $out(u_{k'})$ has no incoming edge from u_k in S, it holds that $w_{ki} = 0$. In order for $u_i \in out(u_{k'})$ to have an incoming edge with $w_{ki} > 0$ from node u_k in S, it should also hold that $u_i \in out(S)$.

$$\begin{aligned} &\sum_{u_i \in (out(u_{k'}) \cap out(S))} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &= \sum_{u_i \in out(u_{k'}) \cap out(S))} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \end{aligned} \quad (18)$$

Combining Equations 17 and 18 and replacing them in Equation 16

$$\begin{aligned} \sigma_j(S \cup u_{k'}) &= \sum_{u_i \in out(S)} \sum_{u_k \in S} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &+ \sum_{u_i \in out(u_{k'})} w_{k'i} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &+ \sum_{u_i \in (out(S) \cap out(u_{k'}))} \sum_{\substack{u_k \in \\ (S \cup u_{k'})}} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \\ &- \sum_{u_i \in (out(S) \cap out(u_{k'}))} \sum_{u_k \in (S \cup u_{k'})} w_{ki} \sum_{l=1}^{n} c_{lj} u_{kl}(t-1) \end{aligned}$$

$$\sigma_j(S \cup u_{k'}) = \sigma_j(S) + \sigma_j(u_{k'}) \quad (19)$$

Similarly, for the seed set $T \cup u_k$ it can be found that

$$\sigma_j(T \cup u_{k'}) = \sigma_j(T) + \sigma_j(u_{k'}) \quad (20)$$

Using Equations 19 and 20 to inequality 13, we conclude that

$$\begin{aligned} \sigma_j(S \cup u_{k'}) - \sigma_j(S) &\leq \sigma_j(T \cup u_{k'}) - \sigma_j(T) \\ \sigma_j(u_{k'}) &\leq \sigma_j(u_{k'}) \end{aligned}$$

∎

REFERENCES

[1] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the 9th ACM SIGKDD*, 2003.

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

[3] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *Proceedings of the 2012 IEEE 12th ICDM*, 2012.

[4] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD*, 2014.

[5] S. A. Myers and J. Leskovec, "Clash of the contagions: Cooperation and competition in information diffusion," in *12th IEEE International Conference on Data Mining, ICDM 2012*, 2012.

[6] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *Proceedings of the 3rd WINE*, 2007.

[7] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *Proceedings of the 6th WINE*, 2010.

[8] N. Pathak, A. Banerjee, and J. Srivastava, "A generalized linear threshold model for multiple cascades," in *Proceedings of the 2010 IEEE ICDM*, 2010.

[9] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th International Conference on WWW*, 2011.

[10] A. Srivastava, C. Chelmis, and V. K. Prasanna, "Social influence computation and maximization in signed networks with competing cascades," in *Proceedings of the 2015 IEEE/ACM ASONAM*, 2015.

[11] I. Litou, V. Kalogeraki, I. Katakis, and D. Gunopulos, "Real-time and cost-effective limitation of misinformation propagation," in *IEEE 17th International Conference on Mobile Data Management, MDM 2016, Porto, Portugal, June 13-16, 2016*, 2016.

[12] S. Datta, A. Majumder, and N. Shrivastava, "Viral marketing for multiple products," in *2010 IEEE International Conference on Data Mining*, 2010.

[13] R. Narayanam and A. A. Nanavati, "Design of viral marketing strategies for product cross-sell through social networks," *Knowl. Inf. Syst.*, vol. 39, no. 3, Jun. 2014.

[14] W. Lu, W. Chen, and L. V. S. Lakshmanan, "From competition to complementarity: Comparative influence diffusion and maximization," *Proc. VLDB Endow.*, vol. 9, no. 2, Oct. 2015.

[15] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD*, 2015.

[16] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10, 2010.

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

[18] V. Levorato and C. Petermann, "Detection of communities in directed networks based on strongly p-connected components," in *2011 International Conference on Computational Aspects of Social Networks (CASoN)*, Oct 2011, pp. 211–216.

[19] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," vol. 1, no. 1, Mar. 2007.

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

[21] G. Valkanas and D. Gunopulos, "Location extraction from social networks with commodity software and online data," in *ICDMW*, 2012.

[22] J. Kim, S. K. Kim, and H. Yu, "Scalable and parallelizable processing of influence maximization for large-scale social networks?" in *2013 IEEE 29th ICDE*, 2013.

[23] W. Mei and F. Bullo, "Modeling and analysis of competitive propagation with social conversion," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 2014.

[24] Y. Zhu, D. Li, H. Guo, and R. Pamula, "New competitive influence propagation models in social networks," in *Mobile Ad-hoc and Sensor Networks (MSN), 2014 10th International Conference on*, 2014.

[25] Y. Zhang, C. Tang, and L. Weigang, "Cooperative and competitive dynamics model for information propagation in online social networks," *Journal of Applied Mathematics*, vol. 2014, 2014.

[26] W. Lu, F. Bonchi, A. Goyal, and L. V. Lakshmanan, "The bang for the buck: Fair competitive viral marketing from the host perspective," in *Proceedings of the 19th ACM SIGKDD*, 2013.

[27] H. J. Hung, D. N. Yang, and W. C. Lee, "Routing and scheduling of social influence diffusion in online social networks," in *2016 IEEE 36th ICDCS*, 2016.

[28] X. He and D. Kempe, "Robust influence maximization," in *Proceedings of the 22Nd ACM SIGKDD*, 2016.

[29] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *Proceedings of the 22Nd ACM SIGKDD*, 2016.

[30] H.-J. Hung, H.-H. Shuai, D.-N. Yang, L.-H. Huang, W.-C. Lee, J. Pei, and M.-S. Chen, "When social influence meets item inference," in *Proceedings of the 22Nd ACM SIGKDD*, 2016.

[31] M. Bressan, S. Leucci, A. Panconesi, P. Raghavan, and E. Terolli, "The limits of popularity-based recommendations, and the role of social ties," in *Proceedings of the 22Nd ACM SIGKDD*, 2016.

[32] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-i. Kawarabayashi, "Dynamic influence analysis in evolving networks," *Proc. VLDB Endow.*, vol. 9, no. 12, 2016.

[33] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun, "Who should share what?: Item-level social influence prediction for users and posts ranking," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11, 2011, pp. 185–194.