Journal of Information Systems and Telecommunication
Vol.11, No.4, October-December 2023, 307-319

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Inferring Diffusion Network from Information Cascades using Transitive Influence

Mehdi Emadi[1*], Maseud Rahgozar[2*] , Farhad Oroumchian[3]

[1].Faculty of Electrical & Computer Engineering, Babol Noshirvani University of Technology, Babol, Mazandaran, Iran.
[2].School of Electrical & Computer Engineering, University College of Engineering, University of Tehran, Tehran, Iran.
[3].Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai, UAE.

Received: 05 Feb 2022/ Revised: 19 Nov 2022/ Accepted: 05 Feb 2023

Abstract

Nowadays, online social networks have a great impact on people's life and how they interact. News, sentiment, rumors, and fashion, like contagious diseases, are propagated through online social networks. When information is transmitted from one person to another in a social network, a diffusion process occurs. Each node of a network that participates in the diffusion process leaves some effects on this process, such as its transmission time. In most cases, despite the visibility of such effects of diffusion process, the structure of the network is unknown. Knowing the structure of a social network is essential for many research studies such as: such as community detection, expert finding, influence maximization, information diffusion, sentiment propagation, immunization against rumors, etc. Hence, inferring diffusion network and studying the behavior of the inferred network are considered to be important issues in social network researches. In recent years, various methods have been proposed for inferring a diffusion network. A wide range of proposed models, named parametric models, assume that the pattern of the propagation process follows a particular distribution. What's happening in the real world is very complicated and cannot easily be modeled with parametric models. Also, the models provided for large volumes of data do not have the required performance due to their high execution time. However, in this article, a nonparametric model is proposed that infers the underlying diffusion network. In the proposed model, all potential edges between the network nodes are identified using a similarity-based link prediction method. Then, a fast algorithm for graph pruning is used to reduce the number of edges. The proposed algorithm uses the transitive influence principle in social networks. The time complexity order of the proposed method is O(n^3). This method was evaluated for both synthesized and real datasets. Comparison of the proposed method with state-of-the-art on different network types and various models of information cascades show that the model performs better precision and decreases the execution time too.

Keywords: Transitive Influence; Network Inferring; Diffusion Network; link Prediction, Random Network.

1- Introduction

Nowadays, online social networks play an undeniable role in propagating information. People are capable of creating contents on a medium to influence other people's opinions. With the increasing importance of online social networks, researchers have been interested in social network analysis. Several methods have been introduced for studying social network behavior on different topics such as information diffusion [1], community detection [2] - [4] link prediction [5] - [8] influence maximization [9], sentiment analysis [10], and expert finding [11]. News, sentiment, rumors, ideas, innovations, and knowledge diffuse over social networks as different types of information. Hence, modeling information diffusion network for any type of information lets researchers apply various social network analysis methods to understand the behavior of people for further social studies. Knowing that information spreads over an underlying network, information diffusion is modeled as a graph in which people are the nodes and the relations between them are the edges.

Like contagious diseases, a diffusion, also called a contagion, occurs when a piece of information is transmitted from one node to another through the edges between them over the underlying network [12]. In this field, any epidemic event disseminated over a social network can be considered as a piece of information. When a member (node) mentions or copies any piece of information from another member (its neighbors), then, it is called to be infected by a contagion. During the process of information diffusion, nodes get infected by a contagion, and an observable footprint is the time of infection. Like

✉ **Maseud Rahgozar**
m.emadi@nit.ac.ir, rahgozar@ut.ac.ir

epidemic diseases, in the outbreak of a disease in a society, the viruses spread from one person to another, while it is unclear by whom each person is really infected. However, the infection time for each person is observable. Similarly, in viral marketing, no one knows who influenced a client, but we know when a client bought the new product.

The main challenge in the field of information diffusion analysis is the lack of knowledge on the structure of underlying network. To study the behavior of people in a social network, the initial requirement is to infer the network structure from the observed data. Inferring the network structure of neurons in neuroscience [13], sentiment in online social networks [14], [15], community detection [16], or the genes in biology [17],[18] are similar points of interest in current researches. The aim of this article is investigating an epidemiology approach to infer the structure of an influence network from a set of information cascades, i.e., the time history of various events occurred in a network.

In recent years some models have been proposed to infer a network from the observed information cascades. In most cases these models try to solve an optimization problem [17]–[20]. This causes a long runtime which is not applicable for real-world networks with a large size. In recent years, with the development of content along with the graph structure, works have placed more emphasis on the use of content. For this reason, less effort has been made to extend pure structure-based algorithms. For example, in the article [21] work is done on the three features: “information, user decision, and social vectors”. In the [22] work is done on the 5 different information source and data mining technique to find hidden influence. In this study[23], yang and friends used the community structure in addition to the information cascade. But in this research, we have worked on pure structure and tried to provide an algorithm for this purpose. For this reason, in order to make a fair comparison, we have compared our work with solutions based on pure structure. Of course, this method can be used to continue the work in any of the combined works of structure and content.

In this paper, we propose a method for modeling the diffusion which results in inferring the diffusion network. The approach of this method is algorithmic and non-optimization. With the help of link prediction concept and proposing an algorithm for pruning the transitive edges in a graph, a time-efficient method is proposed. First, we look for a formula for modeling the influence of a user on another user. Various formulas are presented based on social rules to find the appropriate one. Experimental results show that one of these formulas is more suitable for modeling. The selected model has a better result based on the f1 measure. These experiments are based on synthesized data. Second, with the use of the appropriate model, the propagation network will be inferred. Approximately, we have an influence rate between each of the two nodes, and a generated graph seems like a complete graph. In the real network, we have a direct edge among the smaller number of users. These additional edges are due to the indirect influence (transitive influence) [24], [25] of individuals on one another. We present a heuristic algorithm that identifies and eliminates indirect influence. This identification is based on a social rule called transitive influence. The time complexity of this algorithm is O(n3) which, in comparison with similar algorithms, has an efficient execution time. The experiments show that the proposed method outperforms several state-of-the-art models in both synthetic and real dataset.

The remainder of the article is organized as follows: In the second section, the problem of the inference of diffusion network is defined, and in the third section, the related and previous works have been reviewed. In the fourth part, the explanation of the proposed method is discussed. Section five shows the results of the experiments and evaluations of the proposed algorithm are investigated. For this reason, we use synthesized and real data sets. And in the last section, we conclude our work.

2- Problem Definition

For modeling a diffusion process, information cascades can be employed. Assume user A has communication with user B in a social network. If user A joins a social campaign, then the effect of this event on user B is joining the campaign as a similar action. The process when a piece of information or an action spread from one node to another over a network generates information cascade. A cascade can be specified as two vectors of T and Q. The vector T = [t1,…,tn] represent a time series of infection times of nodes and the features of the contagion (such as user identification) represented in the vector Q = [q1,…,qn]. For the cascade C(T,Q), there are two assumptions [26]: it’s not obvious which of the nodes is affecting each node, and each node can be affected by many nodes.

Consider a hidden network with graph G’ wherein multiple cascades have been spread over that. The main effort is finding graph G which is an estimation of G’, from the observed cascades. Assume that we have a set of cascades ({C1(T1,Q1), … CN(TN,QN)}); the main problem in “inferring diffusion network” is finding the underlying network which caused these cascades.

3- Related Works

The problem of inferring influence network or modeling information diffusion network may be divided into multiple sub-groups considering several aspects of this problem. For example, in the assumptions of one model, the set of information cascades are fully observed [26], [27], but in some, there are missing data in the cascades [28], or the dynamics of network may change over time in some models [28], [29] whereas the other models assume a time-invariant network [26]. Considering a set of information cascades which are infection time series of a network’s nodes, some models proposed to infer the underlying network over which the information diffuses. As far as finding the best possible graph is NP-Hard, in most cases, these models apply methods like Maximum Likelihood Estimation (MLE) to solve the optimization problem which causes a long runtime. As the output of these models, two main aspects of diffusion network would be characterized: structure of the network and its temporal dynamics [27].

The CONNIE [30] method uses convex programming to learn a network under a randomly uniform distribution of transmission time and recovery time. NETINF [26] considers a static network, and the proposed model uses a tree-shaped graph to infer the relationships between nodes from information cascades. In contrast to NETINF, new models have been proposed which assume the network is not static, and the pathways would change over time, and they are dynamic. The NETRATE [27] method, having a set of cascades, maps the parameters of the transmission rate models for each edge. Based on NETRATE, a new method called INFOPATH [29] was developed. The INFOPATH method calculates a pairwise transmission probability for edges between nodes based on information cascades. Then an optimization problem is formed to select the best edges. The best edges that model information cascades with the least error. With the use of stochastic convex optimization, INFOPATH solved the problem of inferential network inference in less time. In previous methods, the assumption of the homogeneity of the relationship between people in an area was significant. But, the hypothesis of the researchers in MMRATE [31] is that of individuals who are different in different topics. This approach focuses on the multifaceted relationship between the network members. The main focus of TOPIC CASCADE [32] is the prediction of the transmission time of a publication on the network. This method solves, as in previous methods, an optimization algorithm for estimating the parameters of the transmission model. TOPIC CASCADE uses an efficient proximal gradient algorithm based on a block coordinate descent for estimation. Other methods also take into account information from contexts such as text content and individuals. In NIMFC [33], different dimensions of information cascades, including: "time, and topographic characteristics of cascades," "user attributes," and "information content" are used to infer diffusion network.

4- The Proposed Method

In the proposed method, the goal was to find the influence network of the nodes in the input data with some information cascades as input data. Information cascades have the time for the activation of each person.

Accordingly, in this method, formulas for "modeling the impact of individuals on each other" have been presented. Various parameters extracted from the information cascades have been used to express formulas. The parameters that have been extracted from the cascades are the time interval of activation of two people in a cascade $(\Delta t_{ab,c})$, the number of cascades where the person is activated (c_a), and the number of times a person ***“b”*** has been activated after the person ***“a”*** (h_{ab}). In this research, we have tried to provide a model that is general and capable of responding to different types of networks. For this purpose, various models have been presented with different combinations of extracted parameters. Different models were tested in a variety of ways to achieve an acceptable general model. Of course, the models presented are based on the rules governing human relationships in social networks. Information cascades display the time of activation or the participation of a node in a particular publication in the order of the event time.

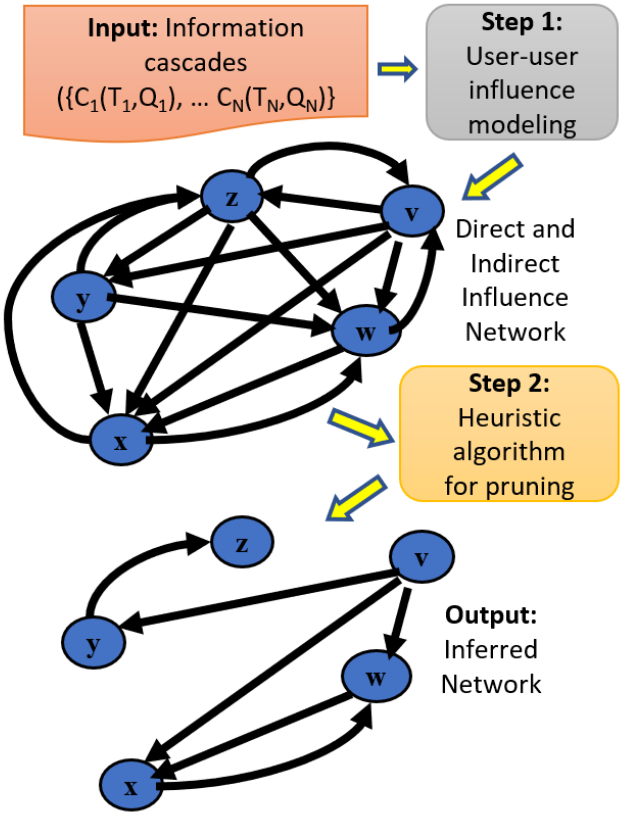

Input: Information cascades ({$C_1(T_1,Q_1)$, … $C_N(T_N,Q_N)$})
Step 1: User-user influence modeling
z
v
y
w
x
Direct and Indirect Influence Network
Step 2: Heuristic algorithm for pruning
z
v
y
w
x
Output: Inferred Network

Fig. 1. The schematic image of the suggested method. With regard to the collection of information cascades, at step 1, the impact of individuals on each other is firstly modeled. This action causes all potential edges between the network nodes to be identified and aggregated in a graph. Then, in step 2, by applying the pruning algorithm on the potential graph, the target graph is deduced.

In this research, by evaluating each cascade, the rate of the influence of the nodes on each other, was calculated. We defined *w*(a,b) as the influence rate of a on b. In each cascade, by observing the activation of b after a, the amount of *w*(a,b) increases. We needed the formula to express how each observation affects the *w*(a,b) calculation. In each information cascade, only the activation time of the node is visible, so the only useful parameter is the activation time. Other parameters can also be used to calculate *w*(a,b), including "the number of caches in which a comes after b," and "the number of cascades where a or b exists."

In Table 1, the parameters extracted from the information cascades have been introduced. "The activation times of a node after another node is activated," "the time interval between the activation of a node with the activation of another node," or "the frequency of activating a node individually" are some of the extracted parameters. In this research, using the same parameters, various models have been provided for calculating *w*(a,b); their list is given in Table 1.

In an output graph between two vertices a and b, where w(a,b) is not zero, we considered an edge (step 1 of Fig. 1). In this way, the output graph had many edges. Most of these edges were derived from our formula of computing the rate of the influence of the nodes on each other(*w*(a,b)), and in fact, we do not have such a direct relationship between the two users.

Table 1. Different functions for different Models of scoring the impact of nodes on each other.

Model	Formula
Model 1 (F. 1) [26]	$\sum \frac{1}{\Delta t}$
Model 2 (F. 2)	$\frac{h_{ab}}{C_a - h_{ba}}$
Model 3 (F. 3)	$\frac{h_{ab}}{C_a - h_{ba}} * \frac{h_{ab}}{C_b}$
Model 4 (F. 4)	$\sum \frac{1}{\Delta t} * \frac{h_{ab}}{C_a - h_{ba}}$
Model 5 (F. 5)	$\frac{h_{ab}}{C_a - h_{ba}} * \frac{h_{ab}}{C_b} * \sum \frac{1}{\Delta t}$
Model 6 (F. 6)	$\sum e^{-\Delta t}$
Model 7 (F. 7)	$\frac{h_{ab}}{C_a - h_{ba}} * \sum e^{-\Delta t}$
Model 8 (F. 8)	$\frac{h_{ab}}{C_a - h_{ba}} * \frac{h_{ab}}{C_b} * \sum e^{-\Delta t}$

With a technique, we must recognize the “real edges” of the “non-real edges”. With the help of the above functions, the influence rate between two nodes is obtained. For some nodes, this number represents the direct effect of these two on each other, which we call “real edge”. But for some nodes, this effect, which has been seen many times in cascades, is due to the indirect effect of two nodes. These edges are called “non-real edge”, which is the result of “Transitive Influence” (Step 2 of **Fig. 1**). For this purpose, a heuristic derived from the social networking space was used. To this end, we tried to find the edges of the transitive influence. The algorithm presented on this heuristic is explained in the following section.

4-1- Proposed user-user Influence Models

Different Models have been presented to calculate the influence rate of one person on another. In all the previous studies, this rate was calculated during an optimization process. However, in this method, the rate has been calculated by reading the cascade information once. In some ways, the method was taken from an optimization problem toward a simple modeling problem and solves the problem in that space.

In most of the conducted researches, they consider Model 1 [26], which is a simple time-based model. And because they raise an optimization problem based on this, they don't need another model. But in this research, we want to determine transitive influence. For this issue, it was necessary to develop and examine different models.

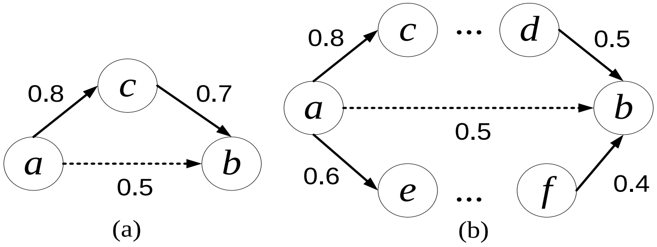


Fig. 2. Transitive influence: (a) node c is affected by node a by 0.8, and node b from node c is affected by 0.7. influence of node c on node b and node a on c causes node a to have an indirect influence on node b of 0.5. (b) As in part a, node b is indirectly affected by node a.

In Table 1., Model 1 and 6 consider the effect of the parameter of the time interval of activation of node b after node a in cascade C in two ways. This formula has been defined exponentially in model 6 for exponential waiting time models and power-law in function 1 for power-law waiting time models. The function 2 defined as division of h_{ab} by $(C_a - h_{ba})$; in other words, "the number of times b is after a" divided by "the number of times b could have come after a." Functions 4 and 7 have been constructed from the combination of functions 1 and 6 with function 2. The simultaneous effect of these two types of functions has been considered in functions 4 and 7. The function 3 multiplies the net effect of b on a in the overall coefficient of influence b to calculate a more normal value than that of function 2. Functions 5 and 8 have been constructed from the combination of functions 1 and 6 with function 3.

4-2- Algorithm for Network Inference

With the aid of the user-user influence model, for all possible edges, the edge weight was calculated. In fact, we obtained a weighted graph close to the complete graph. The weighted graph obtained by the scoring function could not be considered as the actual graph of diffusion network because many of the edges of this graph were derived from indirect influence. But, our goal was to find the direct impact of nodes from each other.

For this purpose, an algorithm was proposed for pruning the indirect edges of the graph and reaching the direct influence graph. For example, in Fig. 2(a), the weight of the edges between the three nodes a, b, and c is calculated using the scoring function. The weight of the edge (a, c) is 0.8, that of the edge (c, b) is 0.7, and that of the edge (a, b) is 0.5. Node b is influenced more by node c and less influenced by node a. Also, node c itself is influenced by node a. The weight of the edge (a,b) is less than these two other edges. According to the indirect influence principle, this edge is due to the indirect influence node a on node b, and it is likely to be said that there is no direct influence between node a and node b. In fact, b through c is influenced by a. So, we can remove this edge.

The indirect influence does not always occur at a distance as large as a node. The distance between two individuals who accept the indirect influence can be more than one node (Fig. 2(b)). In this case, we define a f(a,b) parameter to calculate the rate of indirect influence. In line 7 of algorithm 1 (Fig. 3.), parameter f(a,b) is the maximum flow between the edge a and edge b in the graph G(V, E-(a,b)). If f(a,b) is larger than w(a,b), straight edge (a,b) has been achieved on the basis of indirect influence and should be eliminated (line 8-10 of algorithm 1).

Table 2. Table of Notations for Proposed Algorithm.

G	Graph of user of social network and their possible interaction in all information casscades
E	List of possible interaction between users of Social Network (Edge List)
V	List of users of Social Network (Vertex List)
W	Weight list of extracteg graph from Step 1
w(a,b)	Capacity of edge (a,b) derived from Step 1 of proposed method based on formulas of Table 1.
E’	List of interaction between users of Social Network after graph prunning
W’	Weight list of extracteg graph after graph prunning

According to the selected function, indirect influence is smaller than direct influence. For pruning the edges, we start from the edges with high-weight. If we start with light-weight edges, the algorithm does not work properly.

```
1  input: G(V,E,W)
2  output: G`(V,E`,W`)
3  for all (a,b) ∈ E w’(a,b) ←0 //use w’ as capacity of edges
4  E`←{}
5  sort the edges of E into decreasing order by weight w
6  for each (a,b) ∈ E, taken in decreasing order by weight
7      f(a,b) = FindMaxFlow(a,b, G`(V`,E`,W`))
8      if w(a,b) > f(a,b) then:
9   E` ← E` ∪ (a,b)
10  w’(a,b) = w(a,b)
11  return G`(V`,E`,W`)
```

Fig. 3. Algorithm 1: Pruning the Indirect Edges

4-3- Efficient Algorithm for Pruning Phase

Runtime of the algorithm 1 was not good. We needed a faster algorithm (algorithm 2 in Fig. 4. In this algorithm, the edges of E were sorted in graph G into decreasing order by weight w (Line 5). Then, in line 6, we started with the maximum weight edge. If a path from node a to node b was not available in the graph G', we added the edge (a,b) to the graph G' (Lines 7-8). The findPath(G,a,b) function in this algorithm is a Boolean function that returns true if there is a path from node a to node b in graph G. The algorithm 2 had a better order in terms of time complexity than algorithm 1. Of course, with the help of various experiments, it was shown that they return the same results.

5- Analysis of Algorithms

In algorithm 1, the order of execution for the sorting (line 5 of algorithm 1 in Fig. 3) is equal to O(|E| log |E|). For the second part (lines 6-10 of Fig. 3.) of this algorithm, we can use the Ford–Fulkerson algorithm [34] to calculate the maximum flow. The running time of the Ford–Fulkerson algorithm is equal to O (|E’| max |f|). As a result, the total running time of the second part is equal to O(|E| (|E’| max |f|)).

If we use the Edmonds–Karp algorithm [34] to calculate the maximum flow, the total running time of the second part is equal to $O(|E|(|V'|+|E'|^2))$. Because the running time of the Edmonds–Karp algorithm is equal to $O(|V|+|E|^2)$, the total running time of proposed algorithm is $O((|E| \log|E|+|E|(|V'|+|E'|^2))$. |V|=|V’|. The graph G' is very close to the tree, and consequently, the size of |E’| is equal to c|V|. The size of |E| is equal to $|V|^2$ because G is very close to the full graph. After replacing the new value in the formula, we have $O(|V|^2\log|V|+|V|^4)$. Finally, we have a total time complexity of $O(n^4)$ for algorithm 1. For algorithm 2, we have sorting section (line 5 of algorithm 2 in Fig. 4) too. For the second section of algorithm 2, we have O(|V'|+|E'|) for finding the path method, and the running time of the second part is O(|E|(|V'|+|E'|)). The total running time of the algorithm 2 is O((|E| log|E|+|E|(|V'|+|E'|)). After replacing a new value to the formula, we have $O(|V|^2 \log|V|+|V|^3)$. Finally, we have the total time complexity of $O(n^3)$ for algorithm 2. In the article on INFOPATH [29], there are no references to running time of its algorithm. But our experiments show that the running time of INFOPATH was longer than that of the proposed algorithm in this research.

```
input: G(V,E,W)
output: G'(V,E',W')
W'←W
E'←{}
sort the edges of E into decreasing order by weight w
for all (a,b) ∈ E do:
        if NOT findPath(G,a,b) Then:
                $E' \leftarrow E' \cup \{a,b\}$
return G'(V,E',W');
```

Fig. 4. Algorithm 2: Optimization of the Execution of the Algorithm 1

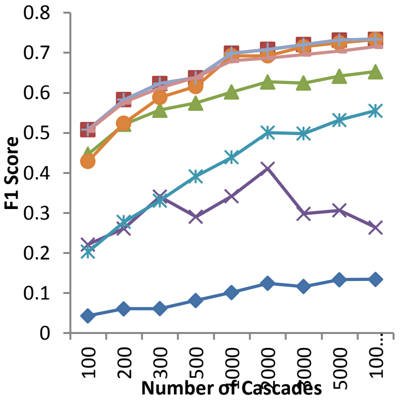
6- Results and Experiments

For the evaluation of our work, we needed data sets to evaluate the proposed method. Due to the inaccessibility of the main graph in this type of problem, synthesized data and real dataset were used to evaluate the proposed methods. Three types of synthesized networks were generated using Kronecker [35] graph models: hierarchical, random, and core-periphery. Also, were generated the information cascades with two types of cascade models: Rayleigh and exponential. For assessment of proposed method with real data, a real dataset from BrightKite social network [36] was used. In this section, we retrieve the graph or network structure by examining the information cascades. In the following, we evaluate the correctness of the algorithm by comparing the resulting graph with the ground truth graph. We used three measures, precision, recall, and f1-score, to evaluate the matching of the network with the main network and to evaluate and compare the methods. The precision is the fraction of inferred edges that are inferred correctly. The recall is the fraction of edges in the ground through network that is inferred correctly. F1-score is computed as the combination of precision and recall (Eq. (1)).

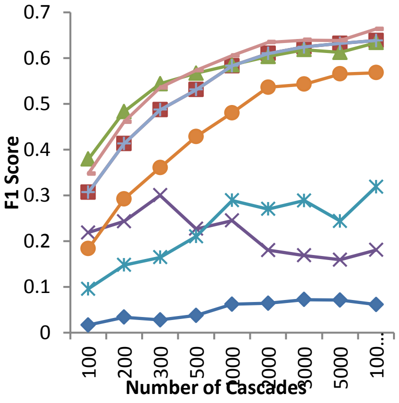
$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

(1)

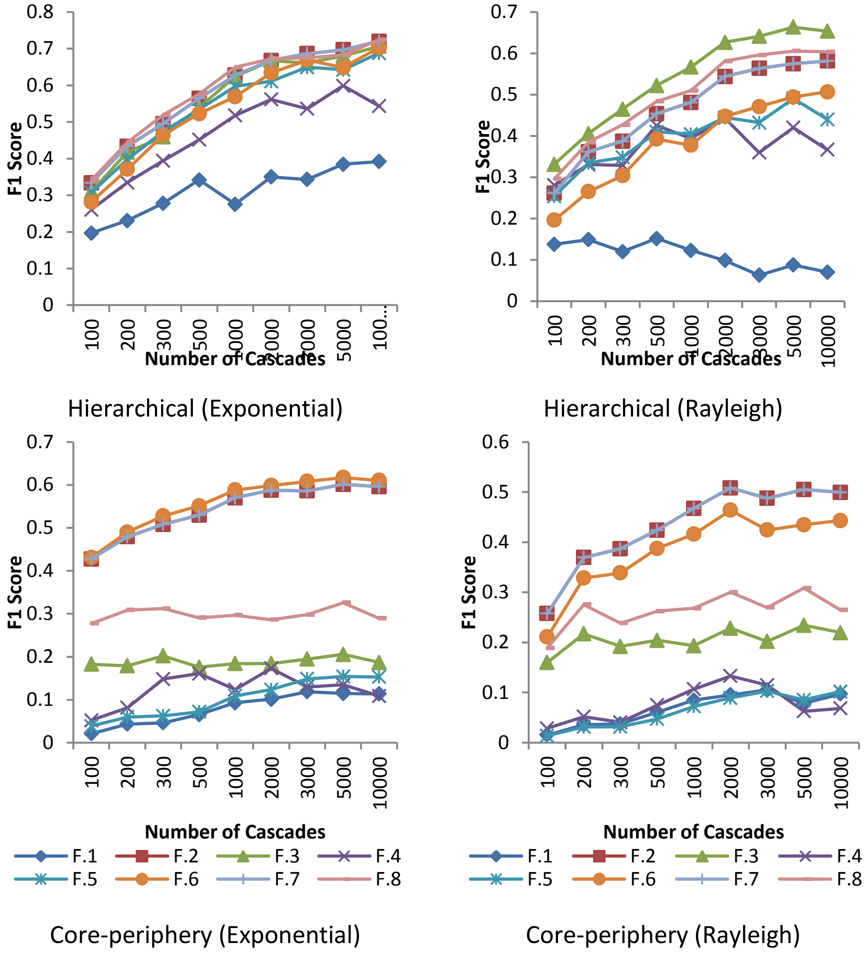
In the remaining sections, we compare the effectiveness of different models in the first subsection, show the experimental results which compare the proposed method with the state-of-the-art method in the next subsection, and in the last subsection, present the result of the experiment on the runtime of the proposed method.



Random (Exponential)



Random (Rayleigh)



Hierarchical (Exponential)

Hierarchical (Rayleigh)

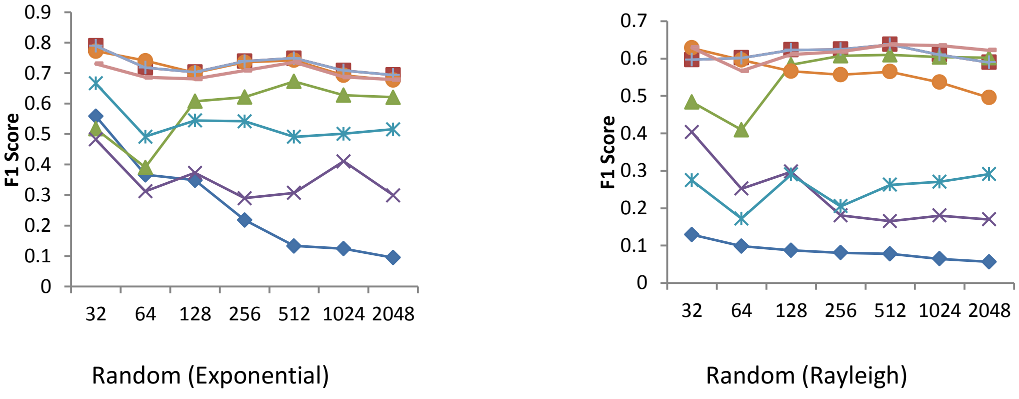
Core-periphery (Exponential)

Core-periphery (Rayleigh)

Fig. 5. Comparing the F1-score of the proposed algorithm with the various user-user influence models (influence rate functions) for the synthesized data with constant network user size (1024 user) and various cascade size from 100 to 10000 .

6-1- Effectiveness of Different Models

By determining the impact value of each node on another node, approximately, we will have the weights for all the edges. The proposed algorithm has been used to find the best influence rate model. Here, we compared all the scoring functions introduced in the previous section using the proposed method and algorithm. To this end, we evaluated various models for different modes of cascade size and network size. For comparison, synthetic data was used. The SNAP tool [37] has been used to produce different types of networks and cascades. Three types of networks (random, hierarchical, and core-Periphery) and two types of cascade models (exponential and Rayleigh) were used for this assessment. For the first mode, the network size was constant, and the number of cascades varied.



Random (Exponential)

Random (Rayleigh)

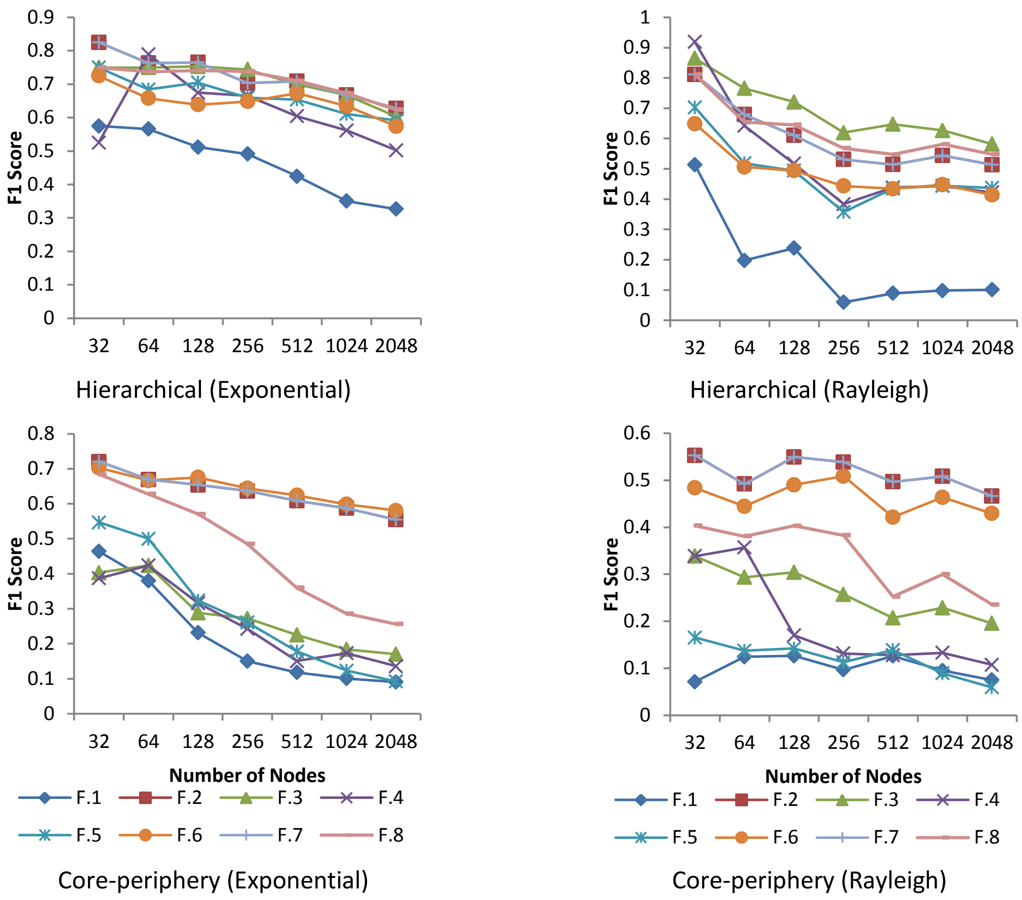
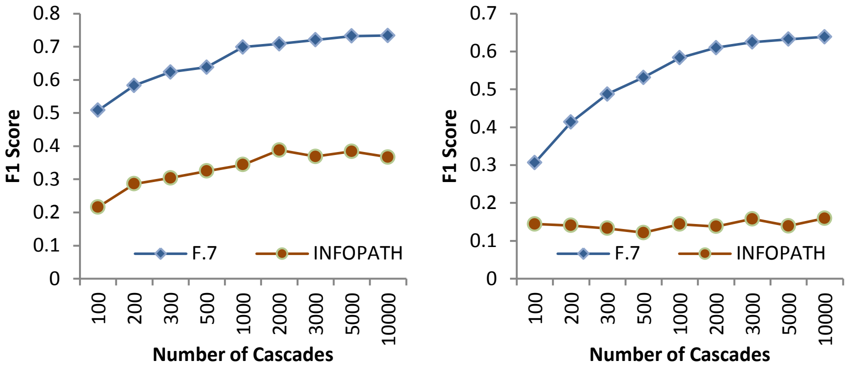
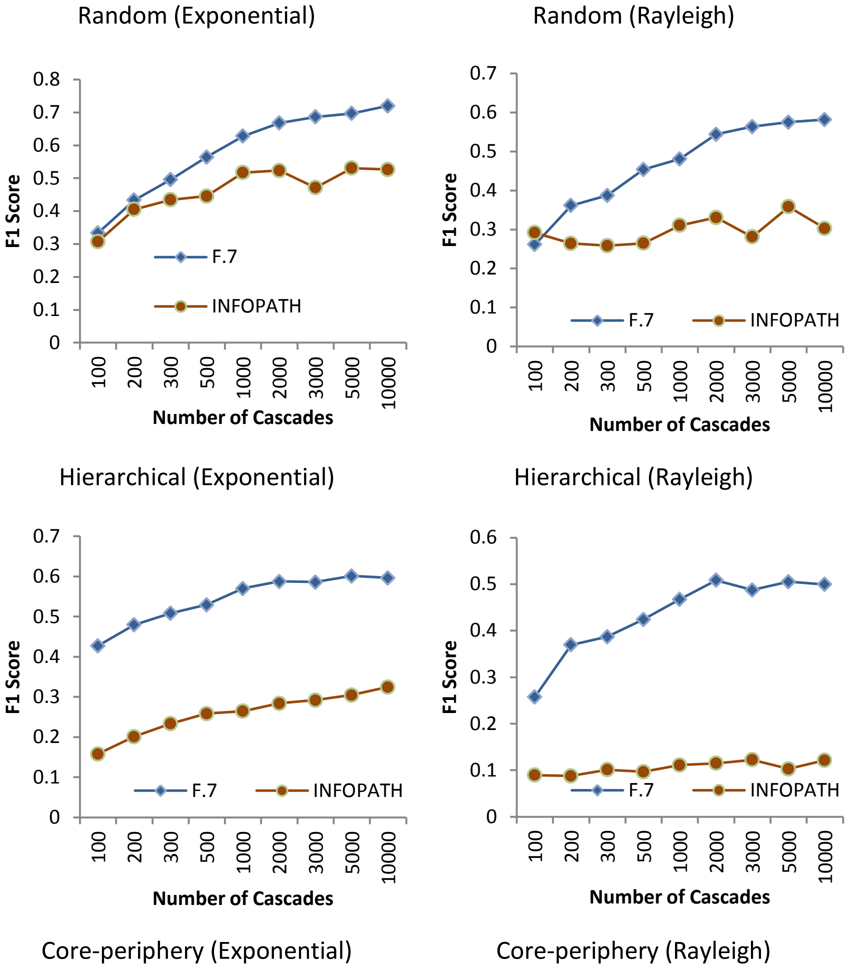


Fig. 6. Comparing f1-score of the proposed algorithm using various user-user influence model (influence rate functions) for synthesized data with constant cascade number (2000 cascades) and various network size from 32 to 2048.

The number of nodes in the network was 1024. The number of cascades varied from 100 to 10000 (100, 200, 300, 500, 100, 2000, 3000, 5000, and 10000). Fig. 5 shows that the increase in the number of information cascades from 100 to 10000 had influenced the performance of the proposed functions. This comparisons show that the proposed functions 3, 7, and 8 were more stable against variation of cascade size and show better performance. By increasing the number of cascades, which is the number of our observations, we will have a better f1measure value. That's why all the diagrams are incremental. Functions 7 and 2 behaved quite similar. This shows that the different parts of the two formulas $(\sum e^{-\Delta t})$ had no effect on performance improvement. Of course, this difference was equivalent to function 6.




Random (Exponential)

Random (Rayleigh)

Hierarchical (Exponential)

Hierarchical (Rayleigh)

Core-periphery (Exponential)

Core-periphery (Rayleigh)

Fig. 7. Comparing f1-score of the proposed algorithm with INFOPATH for the synthesized data with constant network user size (1024 user) and various cascade size from 100 to 10000.

Table 3. Collection of Information Cascades Extracted from the Brightkite Social Network.

Cascade type	How to select users	Number of users	Number of edges	Number of extracted cascades
Type 1	Check in every day	5159	35280	53399
Type 2	Every two days, three times check in	3677	25563	42802
Type 3	Every day twice check in	2805	19871	35826
Type 4	Every two days, five times check in	2241	15690	30386

The value of f1measure for function 6 was also high, and this case shows that function 6 when combined with function 2 does not have much effect on efficiency.

Function 2 has enough information in itself, and adding formula of function 6 will not have much improvement. The function 6 in the eight modes of the nine possible modes of data generated had a lower result, but only in the case of core-periphery exponential, the function 6 had a better answer. Of course, in this case, the function 6 behaved similar to function 7. For other conditions like “core-periphery exponential,” function 3 had a lower f1 value. The function 3 had many oscillations in different states of graph size and cascade numbers, and it was not a good option to choose as the selected model. If we want to choose a function that in most cases is close to the best, we can select the function 7 or 2.

For the second mode, the cascade size was constant, and the network size varied. The number of cascade for the network was 2000. The number of nodes varies from 32 to 2048 (32, 64, 128, 256, 512, 1024, and 2048). For this case, the cascade number was constant, and the number of nodes in the network was changed (Fig. 6); the diagrams have a decreasing behavior. The reason for the downside of the charts was that the number of cascades was constant, and the number of nodes increased. As a result, the ratio of the number of nodes to the number of cascades increased, and the accuracy of the detection of the main edges decreased.

As the charts demonstrated, the behavior of the functions in this mode (variable network size) was the same as in the previous state (variable cascade size). Our experiments show that function 7 gives a better result based on the f1-score measure.

6-2- Comparing the Proposed Methods

Different methods have been proposed to infer the network from information cascades. Each of the methods models a few specific models of networks. The goal of all these methods is to find the best network that models the cascades that have happened. For this reason, their optimization method is sometimes applicable for several specific models of the network or some specific models of cascade production, and they do not do well in the rest of the network. Some methods, such as the INFOPATH, have an acceptable behavior for many types of networks. For the same reason, we compared the proposed method with INFOPATH. To compare our method with the INFOPATH method, we used synthetic and real dataset.

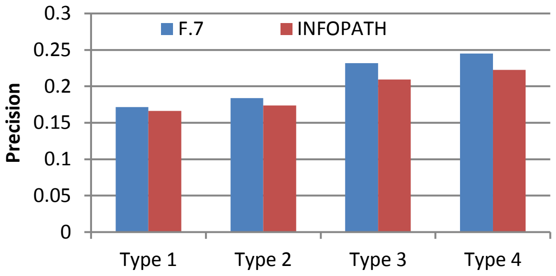


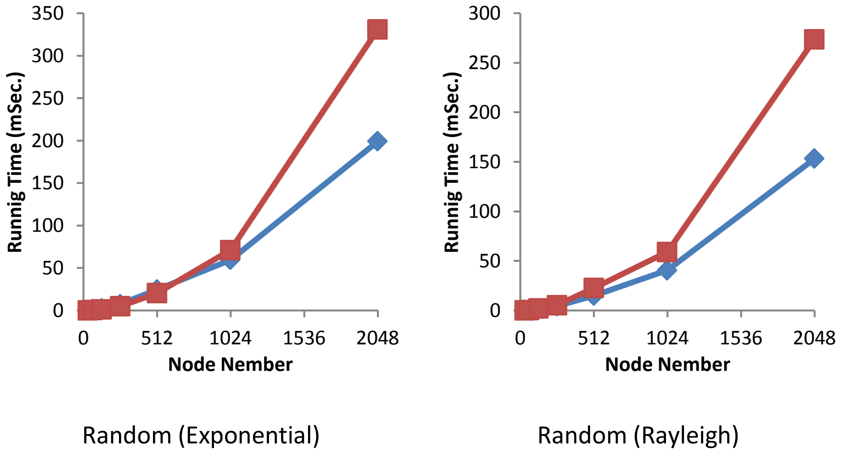
Fig. 8. Precision of proposed method and INFOPATH on BrightKite dataset results for five types of cascades.

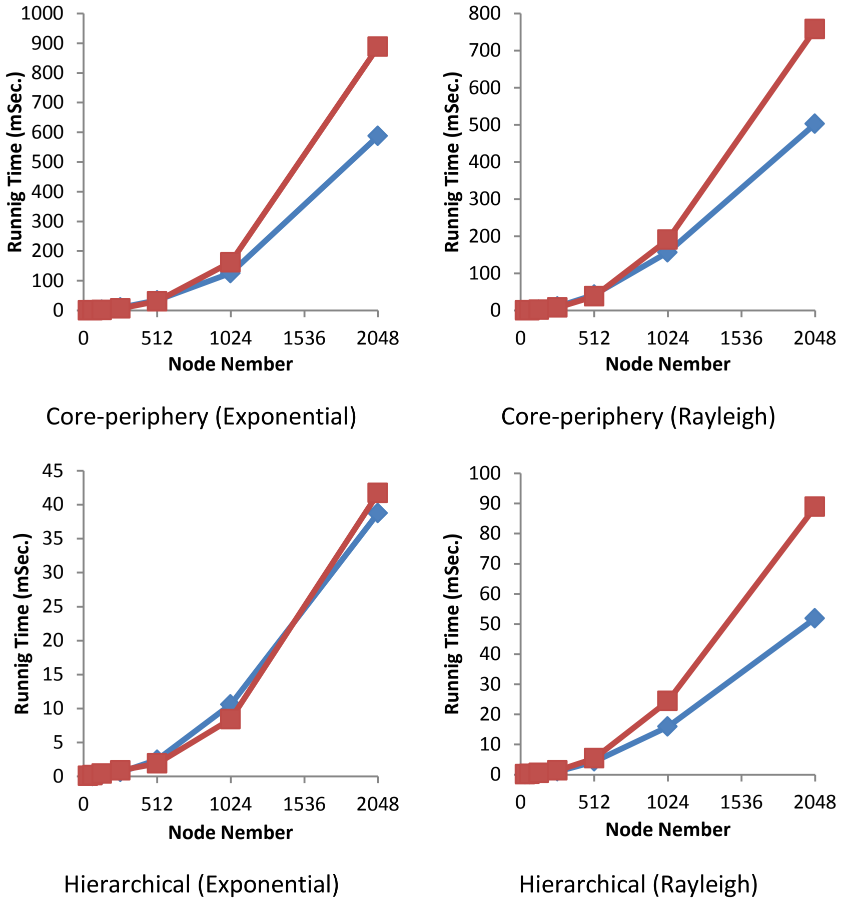
6-2-1 Synthesized Data

To evaluate the proposed method, we produced synthesized data for all the different modes of the network. The SNAP tool [37] has been used to produce different types of networks and cascades. Three types of network (random, hierarchical, and core-periphery) and two types of cascade models (exponential and Rayleigh) were used for this assessment. The number of nodes in the network was 1024. The number of cascades varied from 100 to 10000 (100, 200, 300, 500, 100, 2000, 3000, 5000, and 10000). Fig. 7 shows that the proposed method was better than INFOPATH in term of f1-score measure. The proposed method is highly accurate compared to the INFOPATH for low cascades count. By calculating the average of all execution modes (which are created from the combination of 6 different random network modes, different number of nodes and different cascade sizes), the presented method has improved by an average of nearly 5% based on f1-score measure.

6-1-1 Real Data

The Bright Kite social networking dataset [36] was used to assess the effectiveness of the proposed method on real social networks. In the dataset which is selected from this social network, there were more than 4 million check-ins from over 58000 users, whose relation network is known. In the data-collection, there were 58228 users. There were 214078 communication links between the users. The number of special places according to their unique geographical coordinates was 772966. This dataset was collected from April 2008 to October 2010. In this social network, each person declared his presence after entering a place. Over time, in the profile of each person, the list of places where he or she checked in would be visible.





Core-periphery (Exponential)

Core-periphery (Rayleigh)

Hierarchical (Exponential)

Hierarchical (Rayleigh)

Fig. 9. Comparing running time of the proposed algorithm with INFOPATH for synthesized data with constant cascade number (1000 cascades) and various network size from 32 to 2048 (Redline: INFOPATH;Blueline:F.7).

Also, for each specific location (for example, a restaurant or a cafe), frequent check-ins were recorded from different users. It was necessary to respond to the two challenges, information cascade generation and reducing the size of the datasets, by preprocessing the information to use this dataset to evaluate the proposed method on real social networks.

Checking in the presence of different users for a location is considered as an information cascade. A user will check-in at a specific location (such as a cafe) for the first time, and this will be communicated to his friends on the social network. Then, the friends of that person check-in at that place. So, each information cascade in this dataset is a sequence of checking in for people in a specified location. There are over 700000 unique places in this dataset. But cascade is not made for all of them.

If more than two people are checked in at one place, then a cascade will be created for that place. Due to the large size of this social network, a number of more active users were selected, and their information cascades were extracted (Fig. 8). Four types of cascades were generated. Experiments show that the proposed method had a better precision for all types of the cascade definition (Fig. 8). Averaged over all types of use case, this method provides an improvement of about 2% based on f1-score measure.

6-3- Running Time Analysis

In the previous section, the time complexity analysis of the proposed algorithm was presented. We have shown that the runtime is $O(n^3)$. We have executed proposed algorithm and INFOPATH in the same conditions. The results are showing in Fig. 9. The runtime of INFOPATH was acceptable for smaller values. But by increasing the network volume, INFOPATH runtime increased at high rates. The proposed algorithm for bigger networks has lesser runtime than that for INFOPATH.

Due to the use of stochastic convex optimization to learn the parameters of the information cascade transmission model, the INFOPATH model is relatively faster than the other methods. The time complexity of INFOPATH algorithm is not explained in its article. The experiments to compare the runtime between the INFOPATH model and the proposed method were performed on a machine with 64GB RAM and four processor cores.

6-4- Experimental Environment

All programs are written in C++ language. SNAP library is used for INFOPATH algorithm. The SNAP library has also been used to generate dummy data. All the programs are run in the environment of Ubuntu operating system. The hardware used was a workstation with 32 processing cores and 64 GB of main memory.

7- Conclusion and Future Works

In this research, various functions are proposed to model the impact of individuals on each other in the network. An algorithm based on the indirect influence principle is also presented to infer the graph of the influence of individuals on each other. We evaluated the proposed algorithm based on the various functions affecting the artificial data. As a result of these experiments, we selected model 7 for our proposed method. Then, the proposed method was compared with the INFOPATH method. The INFOPATH model, with hypotheses similar to the proposed method, attempts to infer an influence network based on information cascades. The INFOPATH model has been developed based on the NETRATE model and has been reported to be much faster in terms of runtime. The proposed method has been compared in terms of the accuracy of network inference and runtime with the INFOPATH model on most networks and possible dissemination modes. The comparison of the proposed method with the INFOPATH based on the f1 measure shows that the proposed method can better infer the network. The runtime of the proposed algorithm is of the order of O(n^3). The INFOPATH article does not refer to the time complexity of the algorithm. Therefore, for comparison of these methods, the actual execution time was calculated. From the results of the experiments, the proposed method was found to run faster than the INFOPATH method.

The performed experiments demonstrate that the combination of the time interval and counting parameter creates a better function to calculate the influence rate of individuals on each other. This research only deduces the desired graph based on the information cascades. Therefore, in order to continue to work, information in the content, such as re-tweet or mention, can also be used to improve accuracy. We can also continue to work on functions that improve the performance of the algorithm. Also, specific functions can be provided for real data based on the specific domain of the data.

References

[1] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, “Information Diffusion in Online Social Networks: A Survey,” ACM SIGMOD Record, vol. 42, no. 2, pp. 17–28, 2013.

[2] R. Badie, A. Aleahmad, M. Asadpour, and M. Rahgozar, “An efficient agent-based algorithm for overlapping community detection using nodes’ closeness,” Physica A: Statistical Mechanics and its Applications, vol. 392, no. 20, pp. 5231–5247, Oct. 2013.

[3] Z. Arefian and M.R. Khayyam Bashi. “Scalable Community Detection through Content and Link Analysis in Social Networks,” Journal of Information Systems and Telecommunication (JIST), vol. 4, no.12, pp. 1-10, 2015.

[4] A.H. Hosseinian and V. Baradaran. “A multi-objective multi-agent optimization algorithm for the community detection problem,” Journal of Information Systems and Telecommunication (JIST), vol. 6, no. 3, pp. 166-176, 2018.

[5] E. Sherkat, M. Rahgozar, and M. Asadpour, “Structural link prediction based on ant colony approach in social networks,” Physica A, vol. 419, pp. 80–94, 2015.

[6] V. Martínez, F. Berzal, and J.-C. Cubero, “A Survey of Link Prediction in Complex Networks,” ACM Computing Surveys, vol. 49, no. 4, pp. 1–33, Dec. 2016.

[7] M. P. Salvati, J. Bagherzadeh Mohasefi, and S. Sulaimany. “Overcoming the Link Prediction Limitation in Sparse Networks using Community Detection,” Journal of Information Systems and Telecommunication (JIST), vol. 3, no. 35, pp. 183-190, 2021.

[8] R. Safa, S. A. Mirroshandel, S. Javadi, and M. Azizi. “Publication venue recommendation based on paper title and co-authors network,” Journal of Information Systems and Telecommunication (JIST), vol. 6, no. 21, pp. 33-40, 2018.

[9] K. Rahimkhani, A. Aleahmad, M. Rahgozar, and A. Moeini, “A Fast Algorithm for Finding Most Influential People Based on the Linear,” Expert Systems With Applications, vol. 42, no. 3, pp. 1353–1361, Feb. 2014.

[10] M. Emadi and M. Rahgozar, “Twitter sentiment analysis using fuzzy integral classifier fusion,” Journal of Information Science, vol. 46, no. 2, 2020.

[11] A.A. Kardan and B. Bozorgi. “Analysis of Main Expert-Finding Algorithms in Social Network in Order to Rank the Top Algorithms,” Journal of Information Systems and Telecommunication (JIST), vol. 5, no. 20, pp. 217-224, 2017.

[12] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, “Information Diffusion in Online Social Networks: A Survey,” ACM SIGMOD Record, vol. 42, no. 2, pp. 17–28, 2013.

[13] I. Brugere, B. Gallagher, and T. Y. Berger-Wolf, “Network Structure Inference, A Survey: Motivations, Methods, and Applications,” Oct. 2016.

[14] O. Gomes, “Sentiment cycles in discrete-time homogeneous networks,” Physica A: Statistical Mechanics and its Applications, vol. 428, pp. 224–238, Jun. 2015.

[15] L. Zhao, J. Wang, R. Huang, H. Cui, X. Qiu, and X. Wang, “Sentiment contagion in complex networks,” Physica A: Statistical Mechanics and its Applications, vol. 394, pp. 17–23, Jan. 2014.

[16] L. Prokhorenkova, A. Tikhonov, and Y. Nelly Litvak, “When Less Is More: Systematic Analysis of Cascade-Based Community Detection,” ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 16, no. 4, Jan. 2022.

[17] I. Brugere, B. Gallagher, and T. Y. Berger-Wolf, “Network Structure Inference, A Survey: Motivations, Methods, and Applications,” Oct. 2016.

[18] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, “Inferring Networks of Diffusion and Influence,” ACM Transactions on Knowledge Discovery from Data, vol. 5, no. 4, pp. 1–37, Feb. 2010.

[19] M. Gomez Rodriguez, J. Leskovec, and B. Schölkopf, “Structure and dynamics of information pathways in online media,” in Proceedings of the sixth ACM international conference on Web search and data mining - WSDM ’13, 2013, pp. 23–32.

[20] M. G. RODRIGUEZ, J. LESKOVEC, D. BALDUZZI, and B. SCHÖLKOPF, “Uncovering the structure and temporal dynamics of information propagation,” Network Science, vol. 2, no. 01, pp. 26–65, Apr. 2014.

[21] LiHuacheng, XiaChunhe, WangTianbo, WenSheng, ChenChao, and XiangYang, “Capturing Dynamics of Information Diffusion in SNS: A Survey of Methodology and Techniques,” ACM Computing Surveys (CSUR), vol. 55, no. 1, pp. 1–51, Nov. 2021.

[22] C. Ravazzi, F. Dabbene, C. Lagoa, and A. v. Proskurnikov, “Learning Hidden Influences in Large-Scale Dynamical Social Networks: A Data-Driven Sparsity-Based Approach, in Memory of Roberto Tempo,” IEEE Control Systems, vol. 41, no. 5, pp. 61–103, Oct. 2021.

[23] H. Yang et al., “Towards embedding information diffusion data for understanding big dynamic networks,” Neurocomputing, vol. 466, pp. 265–284, Nov. 2021.

[24] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications, First Ed. Cambridge, United Kingdom: Cambridge University Press, 1994.

[25] K. Chen, P. Luo, and H. Wang, “An influence framework on product word-of-mouth (WoM) measurement,” Information and Management, vol. 54, no. 2, pp. 228–240, Mar. 2017.

[26] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, “Inferring Networks of Diffusion and Influence,” ACM Transactions on Knowledge Discovery from Data, vol. 5, no. 4, pp. 1–37, Feb. 2010.

[27] M. G. RODRIGUEZ, J. LESKOVEC, D. BALDUZZI, and B. SCHÖLKOPF, “Uncovering the structure and temporal dynamics of information propagation,” Network Science, vol. 2, no. 01, pp. 26–65, Apr. 2014.

[28] S. Shaghaghian and M. Coates, “Online Bayesian Inference of Diffusion Networks,” IEEE Transactions on Signal and Information Processing over Networks, vol. 3, no. 3, pp. 500–512, Sep. 2016.

[29] M. Gomez Rodriguez, J. Leskovec, and B. Schölkopf, “Structure and dynamics of information pathways in online media,” in Proceedings of the sixth ACM international conference on Web search and data mining - WSDM ’13, 2013, pp. 23–32.

[30] S. A. S. Myers and J. Leskovec, “On the Convexity of Latent Social Network Inference,” in NIPS’10 Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2, 2010, pp. 1741–1749.

[31] S. Wang, X. Hu, P. S. Yu, and Z. Li, “MMRate: inferring multi-aspect diffusion networks with multi-pattern cascades,” Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’14, pp. 1246–1255, 2014.

[32] N. Du, L. Song, H. Woo, and H. Zha, “Uncover Topic-Sensitive Information Diffusion Networks,” in Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, 2013, vol. 31, pp. 229–237.

[33] D. H. Zhou, W. B. Han, Y. J. Wang, and B. Di Yuan, “Information diffusion network inferring and pathway tracking,” Science China Information Sciences, vol. 58, no. 9, pp. 1–15, Sep. 2015.

[34] C. E. Cormen, Thomas H., Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. MIT Press and McGraw-Hill, 2009.

[35] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, “Kronecker Graphs: An Approach to Modeling Networks,” Journal of Machine Learning Research, vol. 11, no. Feb, pp. 985–1042, 2010.

[36] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility,” in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11, 2011, pp. 1082-1090.

[37] J. Leskovec and R. Sosič, “SNAP: A General-Purpose Network Analysis and Graph-Mining Library,” ACM Transactions on Intelligent Systems and Technology, vol. 8, no. 1, pp. 1–20, Jul. 2016.