DNE: A Method for Extracting Cascaded Diffusion Networks from Social Networks

Motahhare Eslami, Hamid R. Rabiee, and Mostafa Salehi
AICTC Research Center, Department of Computer Engineering
Sharif University of Technology
Email: rabiee@sharif.edu

***Abstract*—The spread of information cascades over social networks forms the diffusion networks. The latent structure of diffusion networks makes the problem of extracting diffusion links difficult. As observing the sources of information is not usually possible, the only available prior knowledge is the infection times of individuals. We confront these challenges by proposing a new method called DNE to extract the diffusion networks by using the time-series data. We model the diffusion process on information networks as a Markov random walk process and develop an algorithm to discover the most probable diffusion links. We validate our model on both synthetic and real data and show the low dependency of our method to the number of transmitting cascades over the underlying networks. Moreover, The proposed model can speed up the extraction process up to 300 times with respect to the existing state of the art method.**

I. INTRODUCTION

In the past decade, complex networks have attracted great attention as an emerging area of multi-disciplinary research. The study of non-trivial topological features of social networks as a type of complex networks has fascinates many network researchers. Information propagation is one of the noticeable topics that has been considered in online social networks. The networks with pieces of information as basic units are called "information networks" [1]. As a piece of information propagates over a network, it leaves a trace from itself that is called "information cascade", and propagating information cascades over an information network build a "diffusion network". Diffusion networks play an important role in social networks analysis. They can be used to study the network structure, link behavior and information propagation paths. Figure 1 illustrates an information network, sample information cascades, and the resulted diffusion network. The diffusion network structure can determine influential individuals and their influence bound that will provide a comprehensive view of individuals' positions and roles in the underlying network [5]. Moreover, inferring the diffusion network can help us to determine the general speed of information propagation in a social network. The structure of diffusion paths also shows the level of information significance by tracking infectious individuals and links.

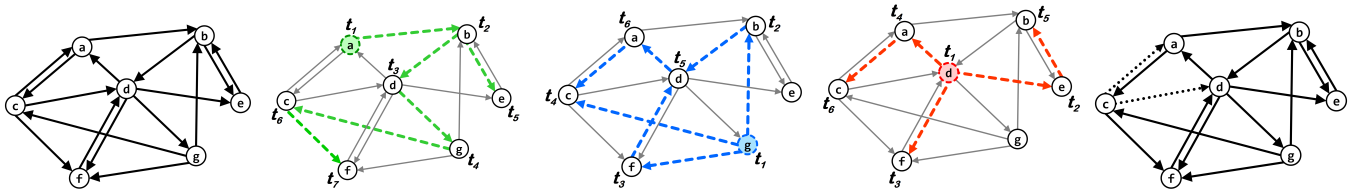
Extracting diffusion networks has many practical applications. Diffusion network inference can be used in ranking strategies for search engines. For example, iRank [2] as a search algorithm, ranks blogs by their role in information propagation. This search engine is designed to find the sources of information (referred to as zero patient) that are important in many applications.

Detecting the links that are connected to spams is another application of diffusion network extraction [3]. The level of information popularity is another parameter that can be analyzed by considering the structure of diffusion networks. Detecting propagation paths can lead to information significance by considering the amount of infected individuals and links. For example, if a government wants to inspect the influence of some news in the society, it can examine the corresponding diffusion networks to gain insight about the popularity and effects of those news.

Diffusion process is a fundamental phenomenon that occurs in many fields. Virus spreading between people, signaling in neural networks and recommending products in viral marketing [11] are some examples of diffusion processes that can take place over various complex networks. In spite of their popularity, the extraction of diffusion networks is a difficult task because of their latent network structure. Fore example, we can usually notice when people receive some information but we do not know who gave them the information. Similarly, in virus diffusion process, the infection phenomenon shows itself when somebody becomes infected but we cannot determine who infected whom [4]. Therefore, inferring the network of diffusion is an ambitious goal when we only observe the infection times.

In the early methods of diffusion network extraction, topological features of network such as node degree, centrality, and betweenness were used to predict the diffusion links [2], [6], [7], [8], [9], [10]. All these works assume that the underlying networks are known and aim to extract the proper diffusion networks. Therefore, they cannot be used in situations without any knowledge about underlying network. Some other approaches have extended the problem and assumed there is no information about the underlying networks structure [4], [5]. These methods try to find the diffusion networks by using the infection times of individuals. However, these methods should propagate many cascades to cover the entire underlying network. Therefore, the accuracy of these methods is very sensitive to the number of information cascades. In addition, the running time of these methods increase very rapidly with the growing size of networks which could be a challenging problem in large scale real networks.

In this work, we propose a scalable algorithm called "DNE"



(a) Underlying information network (b) Cascade 1 (c) Cascade 2 (d) Cascade 3 (e) Diffusion network

Fig. 1. Diffusion process over information networks. (a) Underlying information network. (b,c,d) information cascades propagating over underlying information network. The first source of information spreading is discriminated by a different color. The order of infection times are labeled for the nodes attending in the information cascades. (e) Aggregation of diffusion links of cascades constructs a diffusion network.

(**D**iffusion **N**etwork **E**xtraction) that utilizes the time-series data to extract the diffusion links without assuming any prior knowledge of the underlying network structure and its topological features. We first model the spreading information cascades over the network as a Markov random walk process. Then, we construct an initial underlying network that contains all the probable links attending in the diffusion network. Considering some concepts of random walk such as the hitting time measure, leads us to a new parameter we call the reaching time. The reaching time discriminates the average transmitting steps of infection between two nodes by considering the possible paths between them. As the reaching time for two nodes increases, the probability of infection transmission between these nodes decreases. Hence, we define a probabilistic model for this parameter that leads to a nonlinear computation time. To alleviate this problem, we introduce a tractable approximation that is independent from the cascade transmission model. This independence can be useful in real networks where the actual model of information transmission is not available. We show that our method can run in linear time with respect to the number of the edges in the network. The proposed model (DNE) can speed up the extraction process up to 300 times with respect to the state of the art method introduced in [4].

Experimental evaluations on synthetic and real data show that DNE has the ability to extract diffusion links with little dependency to the number of spreading cascades over the network. Moreover, in some applications the goal is to find the most important diffusion links instead of the entire diffusion network. The proposed method can extract these links with high precision and little dependency to the number of transmitting cascades. Extracting important links in contrast of all diffusion links is beneficial in many situations such as finding influential nodes, critical connections in transmitting information or diseases.

In summary, our main contributions are:

- Speeding up the extraction process to increase the scalability in diffusion network extraction
- Reducing cascade dependency of extracting diffusion links
- Extracting important diffusion links with high precision and low dependency to the number of cascade changes
- Proposing a method that is independent of cascade transmission model

The rest of this paper is organized as follows. In section 2 we provide a classification of the previous work on diffusion network extraction. A formal description of the problem is presented in section 3. The proposed algorithm DNE and experimental results on synthetic and real data are presented in sections 4 and 5, respectively. Finally, the concluding remarks are provided in section 6.

II. Related work

Cascading behavior and diffusion networks have attracted considerable attention in recent years [2], [4], [5], [6], [7], [8], [9], [10], [12], [13], [14], [15]. Although a few studies have explicitly considered the structure of diffusion networks, but there are several lines of research that are related to our work. As there is no extensive investigation on classification of diffusion network extraction, here we try to provide a comprehensive survey on the previous related works. The related research in this field can be classified into three categories:

- *Link Prediction*: The link prediction problem tries to find the future links when the network structure is available as a prior knowledge [16], [17]. Most of the link prediction methods use topological features of network to predict probabilistic connections in the future. The intrinsic difference of link prediction problem with diffusion network extraction is in their goal which are finding probable future links and present diffusion links, respectively. Despite this inherent difference, link prediction methods can be useful in diffusion network inference by considering their approach in predicting the future links. In [2] and [8] the authors have tried to infer diffusion links besides predicting the future links, for the first time.
- *Network Completion*: The network completion problem that is also called network reconstruction, tries to infer the missing parts of the network [18], [19]. At many situations, we can observe only a part of the network and we need to complete its missing parts. Network completion and diffusion network extraction problems have the common goal of finding the missing edges in a network. Although in network completion we have partial knowledge of the network with some topological features,

this information may not be available in the diffusion network inference.

- *Network Inference*: The network inference problem aims to infer the network of diffusion by considering viruses, information, and innovations as the propagation units. The network inference for the first time was studied by [6]. This work tries to reconstruct epidemic trees of disease propagation and estimates the sickness outbreak history. The later work on this topic can be classified into two categories. The first category tries to find propagation links by using the structure and topological features of underlying network [7], [9], [10].
  The second category has a more ambitious goal; it tries to extract the diffusion network without any knowledge of the underlying network. The only available information in this method is the infection times of nodes. Apparently, little work had been done in this category [4], [5]. NETINF [4] is a key work that introduces this category of network inference problem. This method uses an iterative algorithm by considering a sub-modular function to extract the diffusion network. Later on, CONNIE [5] has generalized this concept by removing the homogeneity assumption of the network edges and trying to find the rates of infections in addition of finding diffusion links.

Since our goal is to extract diffusion networks without any knowledge of the underlying network structure, [4] and [5] are the most related works to this paper. Despite admissible accuracy of these algorithms, their running time will grow rapidly by increasing the network size. This growth rate is significant in the large scale networks: therefore the running time must be considered as an intricate factor in diffusion network extraction problems. On the other hand, these methods are highly dependent to the transmitting cascades over the network and decreasing the number of cascades would significantly decrease the network inference precision. Here, we try to extract the diffusion network in a very low running time and decrease the dependency of solution to the number of transmitting cascades.

III. PROBLEM FORMULATION

In this section, we formally describe the diffusion network problem by introducing cascade transmission models and modelling the diffusion process as a Markov random walk.

A. Problem Statement

Consider the directed network $G(V, E)$ with $|V| = n$ nodes and $|E| = e$ edges. Let C denotes the set of cascades that spread over G. Assume C has N_c members with the corresponding time vector $\{t_1, t_2, \cdots, t_n\}$. This time vector shows the infection times of nodes by the cascade. If a node has not been infected within a cascade, its infection time will be set to ∞. As we want to find the diffusion links of a unique propagation process, we generate homogeneous cascades which have the same structure. Since heterogeneity leads to different diffusion behaviours, it is not applicable in the defined problem and will change the goal of it. We wish to extract the diffusion network when the members of C have propagated over the underlying network. Since we do not have any knowledge of the underlying network structure, we can only utilize cascades' time vectors to infer the diffusion network. Therefore, our main goal is to minimize the number of required cascades and running time to achieve high accuracy for extracting diffusion networks.

B. Cascade Transmission Model

Consider the process of information diffusion over a network. When a node is infected by some information, it may decide to pass the information to its neighbour. Usually, the difference between these two infection times will conform to the distribution of t that is called cascade transmission model. In this work, the transmission models are based on the independent cascade model of [20]. In this model, an infected node transmits infection to its neighbours based on predefined density function and transmissions are assumed to be independent. We have used two well-known parametric models; Exponential (EXP) and Power Law model (PL)[4], [3], [5].

In the cascade generation process, a random node starts infecting its neighbours with an average spread rate of β. If the difference between infection times of nodes u and v is defined by $\Delta = t_v - t_u$, the probability of node u infection by node v is given by:

$$P(\Delta) = \begin{cases} e^{-\frac{\Delta}{\alpha}} & \text{EXP} \\ \Delta^{-\alpha} & \text{PL} \end{cases} \quad (1)$$

Where α is an adjustment parameter [4]. Clearly, this probability has an inverse relation with infection time difference of two nodes (Δ). It means when a node gets infected, the longer it remains in the infection state, its tendency to infect others will decrease. Since different cascade structures as various topics of interest may lead to different behaviours, their transmission model will be different too. Our approach will consider this fact by proposing a method which is almost independent from the transmission model. Therefore, it can be used for many different cascade transmission models. We have tried both EXP and PL density functions in our model and obtained similar results. This illustrates that our algorithm is almost independent of the cascade transmission model.

C. Initial Graph Construction

In this step, we present the diffusion network extraction by constructing an initial graph and matching it to a Markov random walk model. Consider the cascade c is transmitting information over the underlying network G. As it spreads over G, it builds a directed path corresponding to the propagation of information. Removing infinity times, we sort the time vector of cascade c denoted as $\{t_{i1}, t_{i2}, \cdots, t_{in}\}_c$. Then we construct an initial graph G_c, which contains all the probable links in the diffusion process. Indeed, each edge (i, j) for which $t_i < t_j$ can be a member of G_c, because any node i can infect node j with some nonzero probability.

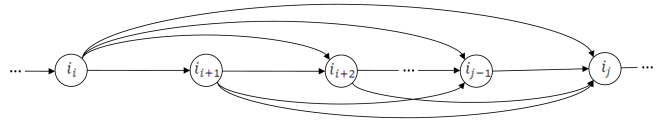
Next, we define a Markov random walk on the nodes of G_c. The infection of nodes, X_t, are defined as the states of the Markov model at time t. For simplicity, we assume at most only one node can become infected in a time step. For each time step, X_t will have a transition matrix P that is associated with links of G_c. This matrix shows probability of infection transmission between any two neighbours. If we define $P(X(t) = j)$ as the probability of node j getting infected at time t, the probability of p_{ij} is given by:

$$p_{ij} = P\{X(t+1) = j | X(t) = i\} \tag{2}$$

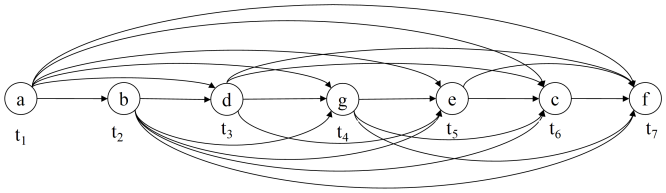
A General state-transition diagram for G_c and the state-transition diagram corresponding to the cascade 1 of Figure 1(b) is shown in Figure 2.

Since we have used the independent cascade model, an infected node infects each of its neighbours independently; therefore infecting a node at time $t+1$ is only dependent to its neighbours getting infected at time t. With this assumption, the diffusion process on the nodes of G_c has the Markov property. That is;

$$\begin{aligned} P\{X(t+1) = x_{t+1} | X(t) = x_t, \cdots, X(2) = x_2, X(1) = x_1\} \\ = P\{X(t+1) = x_{t+1} | X(t) = x_t\} \end{aligned} \tag{3}$$



(a) General initial Graph.



(b) Initial graph of cascade 1 from Figure 1.

Fig. 2. Initial Graph. The nodes are sorted by their infection times and each node is connected to all of the nodes with larger infection times.

D. Reaching Time

We define a parameter called Reaching Time (RT) which is associated with the probability of existence of a link in the diffusion network. This parameter is similar to the hitting time measure in Markov random walk. The hitting time H_{ij} is defined as the expected number of steps before node j is visited, if we start from node i [29]. The hitting time in a strongly connected graph is computed from the following recursive equation [31]:

$$H_{ij} = \begin{cases} 0 & \text{if } i = j, \\ 1 + \sum_{k=0}^{n-1} p_{ik} H_{kj} & \text{if } i \neq j. \end{cases} \tag{4}$$

This equation is obtained by considering the following simple fact. If information wants to go from node i to node j, it should go through one of i's neighbours and continue its path from there. For solving equation (4), we need the stationary distribution of the Markov random walk named π[29]. This distribution can be calculated by using the relation $\pi^T P = \pi^T$ given $\pi^T e = 1$, where e is the standard unit vector.

Equation (4) is useful when our directed graph is strongly connected. If the directed graph of transition probability matrix P is strongly connected, then its Markov random walk will be irreducible and each state is accessible form any other states. The unique stationary distribution of Markov random walk can be obtained if P is irreducible. As G_c is a weakly connected graph, the associated Markov model will not be irreducible and H_{ij} can be ∞ for some pairs of nodes. Therefore, we define the reaching time as a derivative of the hitting time, which measures expected number of steps from node i to node j by considering the *feasible paths*. Feasible paths refer to the paths over which returning to node j from node i is possible. Increasing reaching time between two nodes will decrease the probability of infection transmission between them. Hence, reaching time can be used as a criterion for extracting the most probable diffusion links. We refer to the reaching time for two nodes i and j of G_c as RT^c_{ij}.

E. Diffusion Network Extraction Problem

Now we can define the problem of diffusion network extraction by utilizing the reaching time measure. The inverse relationship between RT and the probability of infection transmission helps us to find links with least RTs as the most likely diffusion links. First, we construct the graph G_{total} as the union of all G_c graphs.

Then we define RT_{ij} for each edge of G_{total} as:

$$RT_{ij} = \sum_{c \in C} RT^c_{ij} \tag{5}$$

Therefore, our problem becomes equal to extracting a subgraph called G' from G_{total}:

$$G' = argmin \sum_{(i,j) \in G_{total}} RT_{ij} \qquad \text{where } |G'| < m \tag{6}$$

Where m is the maximum number of edges in G', and it depends on the properties of the underlying network such as number of nodes and sparsity of the graph.

IV. Proposed Algorithm: DNE

By considering the above problem definition, our goal is to find RTs for each G_c. In this section, we explain the method for one cascade, since the generalization to multiple cascades is straightforward. Intuitively, when some infection spread from node i to node j, it must go to j through those neighbours of i that have a path to j. As Figure 2 shows, the neighbours of node i_i that have a path to node i_j are the nodes whose infection times are between t_{i_i} and t_{i_j} (i.e. $i_{i+1}, \cdots, i_{j-1}, i_j$). Therefore, the recursive equation for obtaining RT is given by:

$$RT_{ij} = \sum_{w=i+1}^{j} p_{iw}(RT_{wj} + 1) \qquad RT_{ii} = 0 \tag{7}$$

As we need to compute the stationary distribution of Markov random walk for non sparse graphs to calculate hitting times, In the best case, calculating all the hitting times for these graphs cannot be less than $O(n^2)$[30]. Since the RT calculations are similar to those of hitting time and G_c is not a sparse graph (with $O(n^2)$ edges), solving this equation is not possible in linear time with respect to the number of edges in G_c. Therefore, we try to find a linear time approximation for RT.

If we assume that the sorted time vector of section III-C has k members, the the node i_j has a link to each of $k-j$ nodes that are being infected after it. Intuitively, we expect each of these links to infect their destinations with a probability of $\frac{1}{k-j}$. However, a more accurate analysis of the problem may lead to a different point of view. Consider node i_1 as the first infected node. By determining the above probabilities, each outgoing link from i_1 has the weight of $\frac{1}{k-1}$. It means that edge (i_1, i_2) also has the probability of $\frac{1}{k-1}$. On the other hand, if we remove any external infection source, the only node that is able to infect i_2 is i_1; this means the probability of edge (i_1, i_2) is equal to 1. Therefore, we have to define the probabilities based on the infected nodes instead of the infecting nodes. Now we can approximate a weight for each edge.

We introduce the set $S_j = \{n_1, n_2, \cdots, n_{k_j}\}$ that includes all the nodes with less infection times with respect to node j; these nodes are sorted by their infection times in an ascending order. Considering the above explanation about the probabilities, it is obvious that the probability of transmitting information by the edge (i, j) is dependent to the size of S_j. It means as the size of S_j increases, the number of likely nodes for infecting node j increases too; Therefore, the probability of attending edge (i, j) in the diffusion network will decrease. This fact results in inverse relation between the probability of edge (i, j) presence in diffusion network and the size of S_j.

On the other hand, we know that each member of S_j has different priority to infect j. Indeed, in our transmission models, when the difference time of infection between two nodes increases, the probability of infection will decrease. Considering this fact, the probability of a link existence between two nodes has inverse relation with the difference times of infection. As we want to propose an independent model to cascade transmission model, we consider the number of infected nodes between two nodes instead of their difference infection times. we define the *Rank* parameter, r_{ij}, for each edge by the inverse relation with the reaching time (For simplicity, here we consider node i as the ith node which is infected in the cascade.):

$$r_{ij} = \begin{cases} 0 & \text{if } i \geq j \\ |S_j| \times (j - i) & \text{if } i < j \end{cases} \tag{8}$$

r_{ij} can be determined as the inverse probability of diffusion link existence between nodes i and j which is an estimation of average number of steps in reaching infection from node i to node j from all feasible paths. Therefore, considering the definition of reaching time, edge rank is an approximation of reaching time which its time complexity is linear with respect to the number of edges in G_c. This will drastically decrease the running time of the proposed method. In last step, we extract m best edges with minimum RTs to construct G' (Algorithm IV.1).

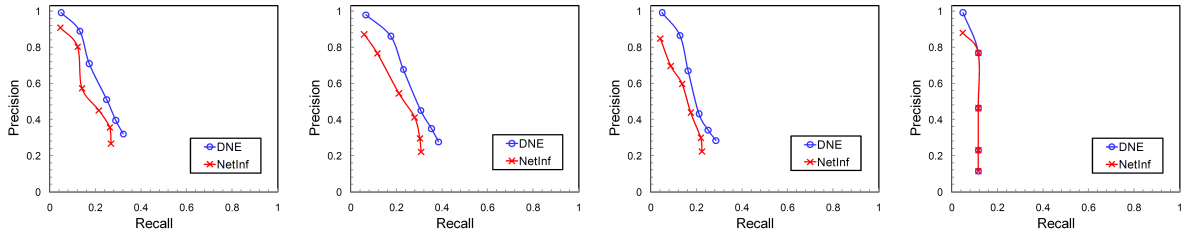
The value of edge rank is not depending on the cascade transmission model as it considers the order of infections instead of the time of infections. Sometimes actual model of information transmission in real networks is not accessible. Therefore, we may use this method for networks with unknown cascade transmission model. In addition, this independency causes DNE to be less sensitive to the changing number of cascades; thus, it can perform relatively good in situations where the number of cascades is small.

Algorithm IV.1: THE DNE ALGORITHM(C, m)

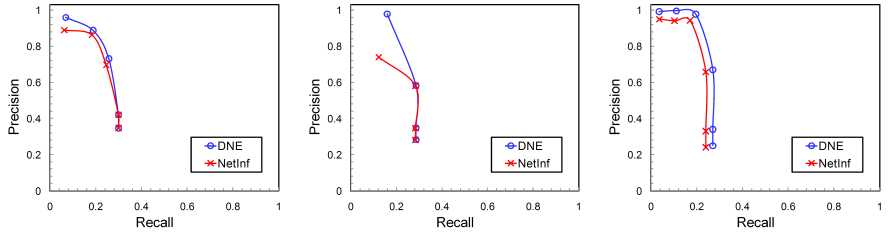
for each $c \in C$
do {
  for each $(i, j) \in c$
  do {
    if $(t_i < t_j)$
    then { $G_c \leftarrow G_c \cup (i, j)$; $S_j^c \leftarrow S_j^c \cup \{i\}$ }
  }
  sort nodes of G_c **by infection time.**
  for each $(i, j) \in G_c$**which** $i < j$
  do {
    $r_{ij}^c \leftarrow |S_j^c| \times (j - i)$
    $G_{total} \leftarrow G_{total} \cup (i, j)$
    $r_{ij} \leftarrow r_{ij} + r_{ij}^c$
  }
}
sort edges of G_{total} **respect to** r_{ij}
for $i \leftarrow 1$ **to** m
  do $G' \leftarrow G' \cup \{e_i \in G_{total}\}$
return (G')

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of DNE for extracting diffusion networks. As it is mentioned in section II, [4], [5] are the most related work to our proposed method. Since CONNIE[5] finds the infectious rate of diffusion links, it is not fair to compare with our method that its main goal is only extracting diffusion network without considering the propagation rates. Additionally, NETINF[4] as the state of the art method of extracting diffusion network performs better in larger networks. Therefore, we compare our algorithm with the well-known NETINF method. Using the order of infection times by DNE results in performing better than NETINF. Although NETINF uses time stamps of the infection, it does not consider the order of them. To illustrate this, consider two pairs of nodes as (i, j) and (k, l). In NETINF, the probability of information transmission between i and j and between k and l will be same if their infection time differences are equal. On the other hand, there may exist different number of nodes with infection times between infection times of the pair (i, j) and the pair (k, l). As it is mentioned in section IV, the number of nodes which are infected between two nodes will affect the probability of the information transmission between these



(a) Random Kronecker (b) Heir. Kronecker (c) Core periphery Kronecker (d) Barabasi-Albert network



(e) Forest Fire network (f) Football network (g) Co-authorship network

Fig. 3. Cascade dependency. Precision and recall for synthetic and real networks by using (a,b,c) 10% (d) 20% (e) 10% (f) 10% (g) 8% of generated cascades.

two nodes while NETINF doesn't consider this fact. Therefore, DNE by applying the order of infections outperforms NETINF in terms of cascade dependency, extracting important diffusion links and running time.

Since the proposed method is independent of cascade transmission model and the results of both transmission models are similar, we only consider the power law transmission model in our simulations.

A. Dataset

To study the effect of underlying network on the performance of DNE, we consider both synthetic and real networks. We use the cascade transmission models (cf. III-B) to generate enough cascades so that 99 percent of edges in the underlying network participate in at least one cascade [4].

1) Synthetic Networks: We use three well-known models for generating directed networks, namely, the Forest Fire model [21], the Kronecker graph model [22] and Barabasi-Albert model [23]. For the Kronecker model, we consider three different networks: Hierarchical [18], Random [24] and Core-Periphery network [25]. Table I provides the parameter matrix of these networks (the same as [4], [5]) and the parameters of cascading process. In the Forest Fire network, parameter matrix contains these parameters: number of starting nodes, forward burning probability, backward burning probability, decay probability and probability of orphan nodes. The parameter matrix for Kronecker graphs is the initial probability matrix and in the Barabasi-Albert model is the number of initial all to all connected nodes.

2) Real Networks: We consider two directed real-world networks. The first network is a co-authorship network of scientists working on network theory and experiment [26]. This network contains 2742 directed links between 1589 scientists. We have generated 6427 cascades over this network.

TABLE I
THE NETWORK GENERATION MODELS' PROPERTIES.

Network Model	Parameter Matrix	α	β	n	e	N_c
Forest Fire	$[5; 0.12; 0.1; 1; 0]$	1	0.5	1024	1221	2786
Hierarchical	$[0.5, 0.5; 0.5, 0.5]$	2	0.5	1024	2048	4000
Random(ER)	$[0.9, 0.1; 0.1, 0.9]$	2	0.4	1024	2048	1813
Core-Periphery	$[0.9, 0.5; 0.5, 0.3]$	2	0.1	1024	2048	2350
BA	$[2]$	2	0.5	1000	2000	4824

Second, we consider the network of American football games between Division IA colleges during regular season fall 2000 [27] which includes 115 nodes and 615 edges with 993 generated cascades.

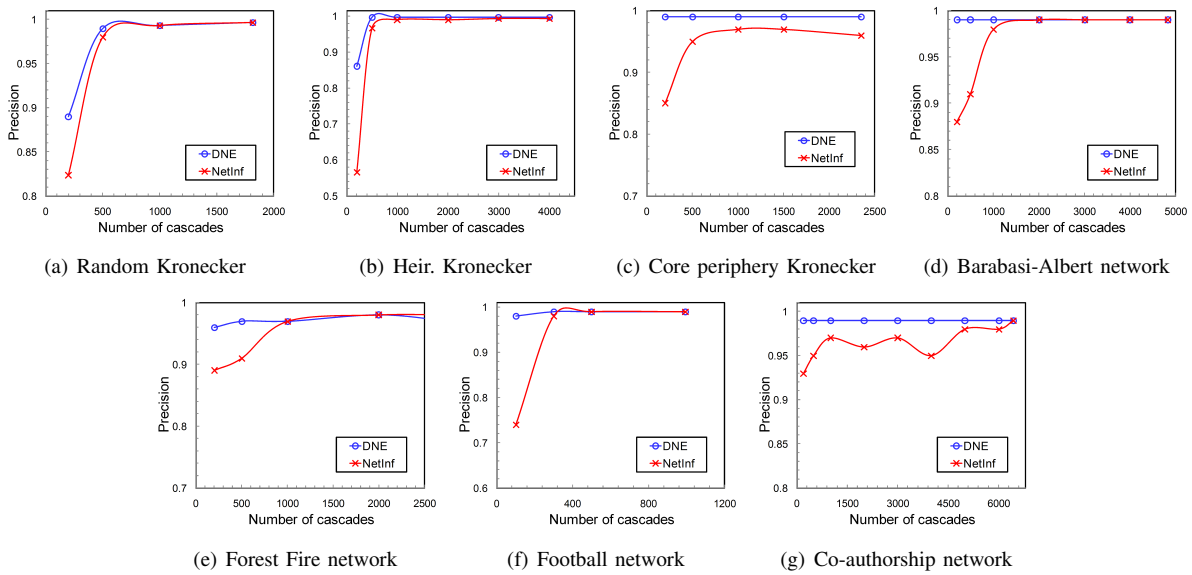
B. Evaluation Metrics

To evaluate the accuracy of DNE, we have measured the precision and recall of our method. Precision refers to the number of correctly inferred diffusion links divided by the total number of inferred diffusion links, and recall refers to the number of correctly diffusion links divided by the total number of links in the network. There is a trade-off between precision and recall; greater precision decreases recall and greater recall leads to lower precision. To consider both precision and recall, we used the F-measure metric. This metric presents a harmonic mean of both precision and recall:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

C. Performance Evaluation

1) Cascade dependency: Obviously, increasing the number of transmitting cascades over the network leads to easier diffusion links extraction. NetInf, for inferring links, at first generates cascades to propagate over the underlying network. For this purpose, we use the same process as NetInf. In NETINF, the number of generated cascades is between 2



(a) Random Kronecker (b) Heir. Kronecker (c) Core periphery Kronecker (d) Barabasi-Albert network

(e) Forest Fire network (f) Football network (g) Co-authorship network

Fig. 4. Extracting important diffusion links. Precision of extracting important diffsion links respect to decreasing number of cascades in (a-e) synthetic networks and (f,g) real networks.

and 5 times more than the number of underlying network edges. As a consequence, using this amount of cascades makes many edges involved in more than one cascade. However, utilizing many cascades is not a realistic assumption. While being independent from the cascade transmission model, DNE reduces the dependency of accuracy to the number of cascades. To measure the level of cascade dependency, we have used about 10 to 20 percent of the number of cascades generated by NETINF (cf. N_c in table I). As shown in Figure 3, DNE outperforms NETINF in terms of precision and recall by about 20% and 10% in average, respectively.

2) Extracting important diffusion links: Some links have a more critical role in propagating information such as borders and links with high load of transmitting information. These links have higher probability in attending in the diffusion network in comparison with other links. In some applications such as finding more potential paths of infection propagation in epidemic diseases, we need to extract the important diffusion links instead of inferring the entire diffusion network.

As shown in Figure 3, DNE efficiently extracts these links with high accuracy and low dependency to the number of cascades. Interestingly, the proposed method has improved the precision by up to 40% in comparison to NETINF.

3) Running time: Since information usually propagates over large scale networks, scalability is an important criterion to extract the diffusion networks. Consequently, in order to improve scalability, we have tried to reduce the running time of diffusion network inference. Without any loss in quality (F-measure), the proposed method drastically decreases the running time in comparison to NETINF (cf. Figure 5: a-g).

In addition, we study the relationship between the running time and network size on a random Kronecker graph. While having the same F-measure, our experiments have shown that DNE performs 300 times faster than NETINF (cf. Figure 5: h).

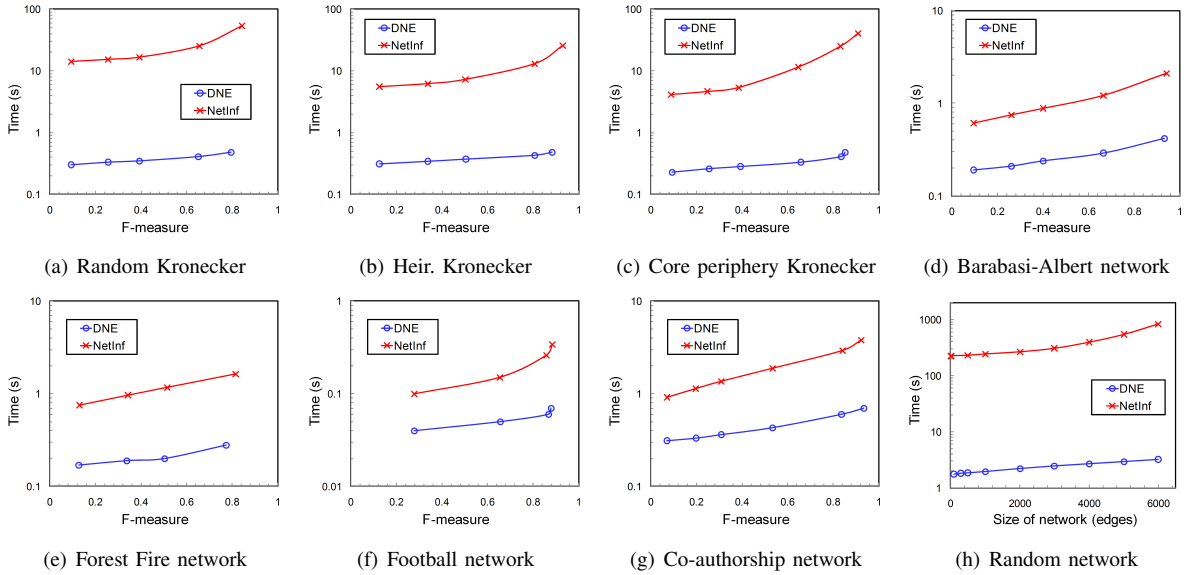
To compare the running time in larger scales, we have used a network that is based on links and posts of blogs in the political blogosphere around the time of the 2004 presidential election in US [28]. It has 1490 blogs and 19090 directed links between them which is covered with 5717 cascades. In this large scale network, while the NETINF method takes 8 hours to infer the diffusion network, DNE infers the diffusion network only in 4 minutes with the same precision.

VI. CONCLUSION

In this paper, we studied the problem of diffusion network extraction by using time-series data without any prior knowledge of the underlying network structure. We proposed a fast and scalable method, called DNE, to infer networks of diffusion. First, we formulated the problem as a Markov random walk on an initial network. Then, we introduced a new metric called reaching time (RT) to extract the most probable diffusion links. We developed an approximation model for reaching time that can be computed in linear time with respect to the number of initial graph edges. Since the proposed method is independent of cascade transmission model, it can perform well when cascade transmission model is unknown.

Evaluating DNE on synthetic and real data showed its low dependency to the number of generated cascades. Moreover, showed that DNE can extract important diffusion links with high precision. Moreover, we showed that our algorithm outperforms the well-known NETINF method [4] in terms of cascade dependency, extracting important diffusion links and running time.

In this paper, we have tried to infer the links of diffusion network. As a future work, it would be interesting to consider time-series data for identifying the speed of information propagation over the social networks.



(a) Random Kronecker (b) Heir. Kronecker (c) Core periphery Kronecker (d) Barabasi-Albert network

(e) Forest Fire network (f) Football network (g) Co-authorship network (h) Random network

Fig. 5. Running Time. Running time of DNE and NETINF in achieving the same F-measure in (a-e) synthetic networks and (f,g) real networks. (h) the relationship between running time and the network size (number of edges) in a random Kronecker graph with 3000 nodes and 6000 edges.

REFERENCES

[1] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, 2010.

[2] E. Adar, L. Zhang, L. Adamic and R.M. Lukose, *Implicit Structure and the Dynamics of Blogspace*, Workshop on the Weblogging Ecosystem, 2004.

[3] J. Leskovec, M. McGlohon, C. Faloutsos, N. S. Glance and M. Hurst, *Patterns of Cascading Behavior in Large Blog Graphs*, In proc. of SDM'07, 2007.

[4] M. Gomez-Rodriguez, J. Leskovec and A. Krause, *Inferring networks of diffusion and influence*, In proc. of KDD '10, pages 1019-1028, 2010.

[5] S.A. Myers and J. Leskovec, *On the Convexity of Latent Social Network Inference*, Advances in Neural Infromation Processing Systems, 2010.

[6] D.T. Haydon, M. Chase-Topping, D.J. Shaw, L. Matthews, JK. Friar, J. Wilesmith, *The construction and analysis of epidemic trees with reference to the 2001 UK foot-and-mouth outbreak*, In proc. of Biol Sci, 270(1511):121-127, 2003.

[7] D. Gruhl, R. Guha, D. Liben-Nowell and A. Tomkins, *Information diffusion through blogspace*, In proc. of of the 13th international conference on World Wide Web, pages 491–501, 2004.

[8] E. Adar and L. A. Adamic, *Tracking Information Epidemics in Blogspace*, Web Intelligence, pages 207–214, 2005.

[9] G. Kossinets, J. M. Kleinberg and D.J. Watts, *The structure of information pathways in a social communication network*, KDD '08, pages 435-443. 2008.

[10] D. Liben-Nowell and J. Kleinberg, *Tracing information flow on a global scale using Internet chain-letter data*, Proc. of the National Academy of Sciences, 105(12):4633-4638, 25 Mar, 2008.

[11] J. Leskovec, A. Singh and J. Kleinberg, *Patterns of Influence in a Recommendation Network*, Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2006.

[12] N. Eagle, A.S. Pentland and D. Lazer, *Inferring friendship network structure by using Mobile Phone Data*, PNAS, pages 15274-15278, 2009.

[13] J. Leskovec, L. Backstrom and J. Kleinberg, *Meme-tracking and the dynamics of the news cycle*, KDD '09: Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 497-506, 2009.

[14] J. Yang and J. Leskovec, *Modeling Information Diffusion in Implicit Networks*, ICDM, IEEE Computer Society, pages 599-608, 2010.

[15] E. Sadikov, M. Medina, J. Leskovec and H. Garcia-Molina, *Correcting for missing data in information cascades*, WSDM, pages 55-64, 2011.

[16] D.L. Nowell and J. Kleinberg, *The link prediction problem for social networks*, CIKM '03: Proc. of the twelfth international conference on Information and knowledge management, pages 556-559, 2003.

[17] B. Taskar, M. Wong, P. Abbeel and Daphne Koller, *Link Prediction in Relational Data*, Advances in Neural Information Processing Systems (NIPS) 16, 2004.

[18] A. Clauset, C. Moore and M. E. J. Newman, *Hierarchical structure and the prediction of missing links in networks*, Nature, 453: pages 98-101, 2008.

[19] M. Kim and J. Leskovec, *The Network Completion Problem: Inferring Missing Nodes and Edges in Networks*, SIAM Conference on Data Mining, 2011.

[20] D. Kempe, J. Kleinberg and E. Tardos, *Maximizing the spread of influence through a social network*, KDD '03: Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, pages 137-146, 2003.

[21] J. Leskovec, J. Kleinberg and C. Faloutsos, *Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations*, KDD '05, 2005.

[22] J. Leskovec and C. Faloutsos, *Scalable modeling of real graphs using Kronecker multiplication*, ICML,ACM International Conference Proceeding Series, 227: pages 497-504, 2007.

[23] A.L. Barabasi and R. Albert, *Emregence of scaling in random networks*, Science, 1999.

[24] P.Erds and A. Rnyi, *On the evolution of random graphs*, Publ. Math. Inst. Hung. Acad. Sci., 5: page 17, 1960.

[25] J. Leskovec, K.J. Lang, A. Dasgupta and M.W. Mahoney, *Statistical properties of community structure in large social and information networks*, WWW, pages 695-704, 2008.

[26] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Preprint physics/0605087, 2006.

[27] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA 99, pages 7821-7826, 2002.

[28] L.A. Adamic and N. Glance, *The political blogosphere and the 2004 US Election*, Proc. of the WWW-2005 Workshop on the Weblogging Ecosystem, 2005.

[29] L. Lovasz, *Random walks on graphs: a survey*, Combinatorics, 2:353398, 1993.

[30] A. Langville and C. Meyer, *Deeper inside pagerank*, Internet Mathematics, 2005.

[31] M. Chen, J. Liu and X. Tang, *Clustering via Random Walk Hitting Time on Directed Graphs*, AAAI Press, 2008.