Inferring Networks and Estimating Influence in Social Media

1. Introduction

The propagation of information, cascading behavior, spread of ideas, viruses and diseases through a network have been studied in different contexts, such as viral marketing (Kempe et al., 2003), the spread of news and opinions (Adar et al., 2004), the spread of technological inovations (Rogers, 1999), and the spread of infectious diseases (Anderson et al., 1992). In such cases, we have a set of contagions diffusing across a network. As a contagion spreads in a network (static or dynamic) it creates a trace at that instant of time, called a cascade. In order to study network diffusion, there are usually two fundamental challenges that need to be addressed. First, to be able to track cascading processes of contagion diffusion, which is certainly nontrivial with large-real world graphs. For example, given data about when did which nodes get infected by a virus, one can construct an exponentially large number of trees containing the infected nodes consistent with the data and the exact inference problem is simply intractable. Second, in a typical scenario only the contamination of nodes as a function of time is observed and the structure of the diffusion network is often unknown or unobserved. For example, in a viral marketing setting, it is easy to track the products purchased by people with respect to time, but not who influenced them into purchasing those.

Another fundamental task, especially with applications in viral marketing (Kempe et al., 2003), is detecting the most (or least) influential nodes in the network. The influence maximization (or minimization) problem can be defined as: find a set of nodes whose initial adoptions of a certain idea or product or contamination can trigger, *in a given time window*, the largest (or smallest) expected number of follow-ups (Du et al., 2013a). A majority of the previous literature have focused on estimating influence for an infinite time window (Kempe et al., 2003; Chen et al., 2009; Goyal et al., 2011). In tasks such as viral marketing, influence at large times is of little use, since typically the marketer would like his advertisement to be viewed in a relatively short period of time (maybe a week or a month).

A series of recent work (Rodriguez et al., 2013; Du et al., 2012; Rodriguez et al., 2011; Du et al., 2013b) in the literature has shown that modeling cascade data and information diffusion through *continuous-time* diffusion networks can provide significantly more accurate networks for inference, than discrete-time models. This modeling choice has two important justifications. First, since a node can be infected at any instant of time, representing time as a continuous variable seems much more appropriate. Also, artifically discretizing time into bins introduces the additional overhead of choosing optimal tuning parameters such as bin size. Second, discrete time models can only describe transmission times which obey an exponential density, and thus are too restricted to model any arbitrary distribution in the data (Du et al., 2013a).

This report develops a method for answering two major questions related to information diffusion: (1) What is the global structure of the underlying network over which information propagates and how do individual contagions cascade over such a network? (2) What nodes are the most/least influential in spreading the contagion? Specifically, to answer the first question we build on the approach by (Gomez Rodriguez et al., 2010) which finds a near-optimal solution for the network graph that maximizes the conditional probability of diffusion of a set of observed contagions given a specific graph (maximum likelihood estimator). We propose an improvement on their model by using the Weibull distribution, a richer family of distributions than the exponential or power law distributions (Lawless, 2011). After the network structure is learned, to address the second question, we build on the approach by (Du et al., 2013a) which estimates and maximizes the influence based on a continous-time diffusion model. Although they achieve state-of-the-art results on both synthetic and real world datasets, their model assumes the transmission times between edges to be independent of each other. We propose a model which incorporates dependencies between transmission times/functions by using priors. Furthemore, we propose a solution to the influence minimization problem as well, which is an important inference problem in applications where identifying non-influential sources might be of great interest, such as viral marketing.

2. Background & Related Work

Generally there are three research topics in information diffusion(Guille et al., 2013): 1) what topics are now popular and diffuse the most in the network, 2) how, and through which paths, information is diffusing, 3) which nodes in the network play important roles in the diffusing process.

We briefly review each of the three topics.

2.1. Detecting Popular Topics

The main tasks here is to automatically detect topics that are popular or will become popular in textual streams, and it has been suggested to focus on bursts. Bursty topic is defined as: A behavior associated to a topic within a time interval in which it has been extensively treated but rarely before and after. (Kleinberg, 2003) proposes a state machine to model the arrival times of documents in a stream to identify bursts, but assume all documents belong to the same topic. (Leskovec et al., 2009) show that the temporal dynamics of the most popular topics in social media are made up of a succession of rising and falling patterns of popularity. (Shamma et al., 2011) propose a Peaky Topics model, which is based on a normalized term frequency metric. They consider each time slice as a pseudo-document composed of all the messages in the corresponding collection. Using this metric, bursty topics defined as single terms are ranked. One problem of this method is that a single term is not enough to clearly identify a topic. (AlSumait et al., 2008) propose a on-line LDA to find latent topics, to train a topic model for current time slice, the topic model for previously generated models is treated as a prior. (Lu & Yang) develop a method that permits predicting which topics will draw attention in the near future. They utilize Moving Average Convergence Divergence (MACD) to identify bursty topics. The main idea is to turn a short period and a longer period moving average of terms frequency into a momentum oscillator. When the trend momentum changes from negative to positive or from negative to positive, the trends of a term will rise or fall. (Takahashi et al., 2011) propose to use mentions contained in messages to identify bursty models, instead of textual content. They combine a mentioning anomaly score and a changepoint detection technique based on Sequentially Discounting Normalized Maximum Likelihood. The anomaly is calculated with respect to the standard mentioning behavior of each user.

2.2. Modeling Information Diffusion

The diffusion process is characterized by two aspects: the graph structure that transcribes who influenced whom, and its temporal dynamics, i.e. the evolution of the diffusion rate which is defined as the amount of nodes that adopts the piece of information over time. We can distinguish two categories of models in the following ways.

2.2.1. Explanatory Models

Explanatory models aim at learning the underlying directed graph structure through which a contagion is diffusing. In such processes, one only observes the time sequence of node infection, but is unable to infer what node was responsible for transmitting the infection. (Gomez Rodriguez et al., 2010) approach that problem by trying to find a near-optimal solution for a probabilistic model of the influence diffusion. They use the Independent Cascade Model (Kempe et al., 2003) and incorporate a time-dependent component to it. The model states that an infected node has a probability β of infecting each one of its neighbour nodes (each infection being an independent process). If such an infection takes place, the infected neighbour will have an incubation time α of the contagion agent before it is able to infect its neighbour nodes. The model is such that the propagation likelihood $P_c(u,v)$ of a contagion c between two nodes u and v is such that $P_c(u,v) = 0$ if $t_u > t_v$ and, otherwise, depends only on the difference between the hit times of the two nodes ($\Delta_{u,v} = t_v - t_u$). They use two distributions,:

- Exponential Model: $P_c(u,v) = P_c(\Delta_{u,v}) \propto e^{-\frac{\Delta_{u,v}}{\alpha}}$
- Power-law Model: $P_c(u,v) = P_c(\Delta_{u,v}) \propto \frac{1}{\Delta_{u,v}^{\alpha}}$

Further on, they use the concept of ϵ-edges to model the influence of factors external to the diffusion network. They aim to solve the optimization problem $G^{\star} = \arg\max_{|G|\leq k} F_C(G)$, where the objective is a submodular monotonic nondecreasing function. They use the popular solution proposed by (Nemhauser et al., 1978) to obtain $1 - 1/e$ optimality guarantee.

2.2.2. Predictive Models

Predictive models aim at how a specific diffusion process would spread in a given network by learning from past diffusion traces. There are generally two models, one is the Independent Cascades (Goldenberg et al., 2001), and the other is Linear Threshold (Granovetter, 1978). The IC model requires a diffusion probability to be associated to each edge whereas LT requires an influence threshold for each node. In IC, for each iteration, the newly activated nodes try once to activate their neighbors with the probability defined on the edge joining them; in LT, for each iteration, the inactive nodes are actived by their activated neighors if the sum of influence degrees exceeds their own influence threshold. (Galuba et al., 2010) propose to use LT model to predict the graph of diffusion. Their model relies on parameters such as information virality, pairwise users degree of influence and user probability of adopting any information. The parameters are optimized by gradient ascent method.

2.3. Identifying Influential Spreaders

Another major task which has numerous potential applications such as in viral marketing, is the task of influence maximization. Influence maximization is defined as finding a set of nodes whose initial adoptions of a certain idea can produce or trigger, in a time window, the largest expected number of follow-ups (Du et al., 2013a). Some approaches have developed discrete-time models (Rodriguez et al., 2013; Du et al., 2012; Rodriguez et al., 2011; Du et al., 2013b), which are typically disadvantageous as they introduce additional tuning parameters such as bin size to discretize time axis into bins and can only model transmission times obeying exponential density. (Du et al., 2013a) introduce a scalable continous-time diffusion model (ConTinEst) with heterogeneous transmission functions to estimate the influence of A sources within a T time window. They use Cohen's modified Dijkstra's algorithm approach (Cohen, 1997) to create least label list for each source node s. Based on the list, they estimate the neighborhood of s within T as,

$$|N(s,T)| \approx \frac{m-1}{\sum_{u=1}^{m} {r_*}^u} \tag{1}$$

where r_* is the least label and m is the number of random label samples. Thereafter, they obtain the neighborhood of A within T by simply taking the union of the neighborhoods of each source. In contrast to the naive sampling approach which needs to run the shortest path algorithm again if the source set is increased, this randomized algorithm involves only a constant-time minimization over $|A|$ numbers. For n samples, the overall computational complexity of the randomized algorithm is given by $O(n|E|log|V| + n|V|log^2|V|)$. They use the Weibull distribution (Lawless, 2011), providing more flexibility than an exponential distribution. They achieve ground truth accuracy on synthetic data (Leskovec et al., 2010) and state-of-the-art accuracy on the MemeTracker dataset (Leskovec et al., 2009).

3. Method

In this section, we present the details of our method organized by: (1) infer the global structure of the underlying network, (2) estimate the influence in the learned network, and (3) optimize the influence. At each step, we use a *continuous-time* diffusion model.

3.1. Static Network Inference

For our purposes, we assume that the underlying diffusion network remains static (does not change over time).

Cascade transmission model. We build on the Independent Cascade Model (Kempe et al., 2003) and the cascade transmission model by (Gomez Rodriguez et al., 2010). The Independent Cascade Model posits that an infected node u infects each of its neighbors in network G independently at random with some probability β. It implicitly assumes that every node can be infected by at most one other node. Therefore, v can have multiples of its nodes infected but only one of those can activate v. Thus, the structure of a cascade c over the graph G, is a directed tree T. In contrast to using the exponential and power-law distributions (Gomez Rodriguez et al., 2010), we define the probabiltiy $P_c(u,v)$ that a node u spreads the cascade to a node v, by a broader (less restrictive) family of distributions known as the Weibull distribution (Lawless, 2011),

$$P_c(u,v) = \frac{\gamma}{\alpha}\left(\frac{\Delta_{u,v}}{\alpha}\right)^{\gamma-1} e^{-\left(\frac{\Delta_{u,v}}{\alpha}\right)^{\gamma}} \tag{2}$$

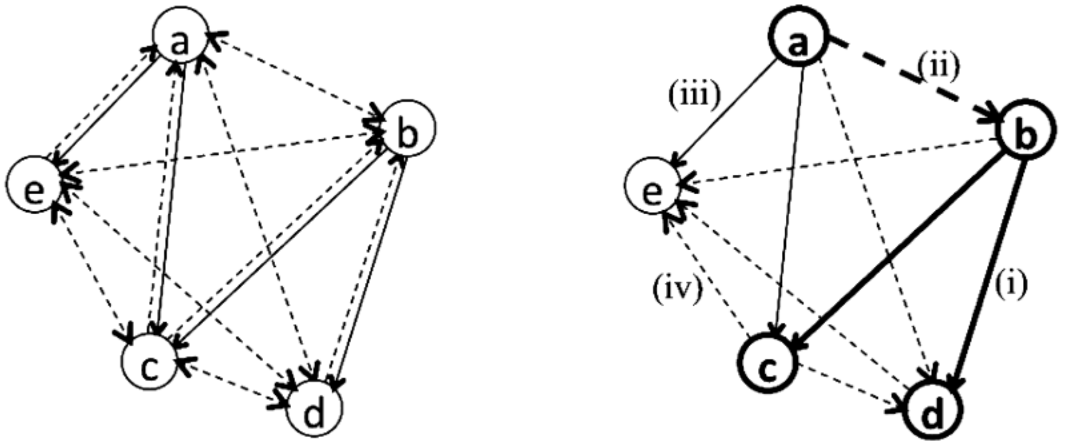
where $\alpha > 0$ and $\gamma > 0$, are distribution parameters corresponding to scale and shape, respectively and $\Delta_{u,v} = t_v - t_u$ is the infection time differential between node u and node v. In large scale networks, most contagions can only infect a small subset of the all the nodes in the network. We set the infection times for uninfected nodes v (after completion of the cascade) to be $t_v = +\infty$ resulting in $P_c(u,v) = 0$. Also, $P_c(u,v) = 0$ if $t_u > t_v$, that is an uninfected node u cannot influence an infected node v.

Many nodes in the network may get infected due to reasons other than the network influence. For example, in viral marketing a person may purchase a product due to TV commericals rather than peers, creating a disconnected cascade (Leskovec et al., 2007). A simple approach to take this into account would be to create an external node x and connect it to every other node u in the network with an ϵ-edge. Then every node u can be infected by the external influence x with a small probability ϵ. However, introducing additional nodes in an already intractable inference task will make our problem even harder. To navigate this, we instead connect nodes u and v by ϵ-edge if they are not connected by a network edge (in that direction) already (Gomez Rodriguez et al., 2010). So, now every node can influence every other node in the network even if they are not connected by a network edge. Our graph G is now a fully connected graph of two disjoint sets of edges, the network edges E and the ϵ-eges E_ϵ, i.e., $E \cap E_\epsilon = \emptyset$ and $E \cup E_\epsilon = V \times V$

Maximum likelihood estimation. Given data of the form $(c, \boldsymbol{t_c})$, where c is the contagion and $\boldsymbol{t_c}$ are the node hit times, we can calculate the likelihood $P(c|T)$ of the contagion c spreading in a particular tree pattern $T(V_T, E_T)$ in the following manner,

$$P(c|T) = \prod_{u \in V_T} \prod_{v \in V} P_c^{'}(u,v) \tag{3}$$

where the transmission probability $P_c^{'}(u,v)$ is given by (Gomez Rodriguez et al., 2010),



(a) Graph G on five vertices and four network edges (solid edges). ε-edges shown as dashed lines

(b) Cascade propagation tree $T = \{(a,b),(b,c),(b,d)\}$

Figure 1. (a) Graph G with network edges (solid) and ϵ-edges (dashed). (b) Propagation tree T = (a,b), (b,c), (b,d) with (i) network edges that are a part of the tree T (solid bold), (ii) ϵ-edges that are a part of T (dashed bold), (iii) network edges that are not a part of T (solid), and (iv) ϵ-edges that are not a part of T (dashed) (Gomez Rodriguez et al., 2010)

$$P_c^{'}(u,v) = \begin{cases} \beta P_c(u,v), & \text{if } t_u < t_v \text{ and } (u,v) \in E_T \cap E \\ \epsilon P_c(u,v), & \text{if } t_u < t_v \text{ and } (u,v) \in E_T \cap E_\epsilon \\ 1-\beta, & \text{if } t_v = \infty \text{ and } (u,v) \in E \setminus E_T \\ 1-\epsilon, & \text{if } t_v = \infty \text{ and } (u,v) \in E_\epsilon \setminus E_T \\ 0, & \text{otherwise (i.e. if } t_u \geq t_v) \end{cases} \tag{4}$$

Figure 1 illustrates the concept of ϵ-edges (Gomez Rodriguez et al., 2010). Figure 1(a) shows an example of the entire graph G, and Figure 1(b) shows the edges participating in the computation of Eq. (4) for the propagation tree $T = (a,b),(b,c),(b,d)$ in graph G. The likelihood $P(c|G)$ of the contagion c for a graph G, can then be calculated by marginalizing over the set of all the directed maximum spanning trees $\mathcal{T}_c(G)$ on a subgraph of G induced by the nodes that got infected by contagion c. Specifically,

$$P(c|G) = = \sum_{T \in \mathcal{T}_c(G)} P(c|T)P(T|G) \tag{5}$$

We assume that the prior probability of a tree T given a graph G follows a uniform distribution, i.e. $P(T|G) = 1/|\mathcal{T}_c(G)|$. Now, the probability of all the cascades for all the set of contagions C is simply given by,

$$P(C|G) = \prod_{c \in C} P(c|G) \tag{6}$$

Taking the log of the likelihood, our maximization problem simply becomes,

$$\hat{G} = \arg\min_G F_C(G) \tag{7}$$

$$= \sum_{c \in C} \max \sum_{(i,j) \in E_T} log(P_c^{'}(i,j) - log(\epsilon P_c(i,j)) \tag{8}$$

where the second log term inside the summation comes from the empty graph K (a graph only with ϵ-edges). Maximizing with respect to K makes the objective in Eq. (8) a monotic nondecreasing submodular function (Gomez Rodriguez et al., 2010). The log differential inside the summation is non-negative and can be thought of as an improvement on the log-likelihood for edge (i,j) under the most likely propagation tree T. Since, Eq. (8) is monotonically nondecreasing, the graph G which maximizes it will be a fully connected (network edges) graph. However, real-world graphs are in general sparse and so we want to learn the graph that only contains some small number of k network edges. Our optimization problem then becomes,

$$G^{\star} = \arg\max_{|G| \leq k} F_C(G) \tag{9}$$

where maximization is over all graphs G containing k edges. Maximizing submodular functions is in general NP-hard (Khuller et al., 1999). However, (Nemhauser et al., 1978) have shown that using the *greedy* approach we can

Algorithm 1 Max. weight directed spanning tree of a DAG

```
**Require:** Weighted DAG $D(V, E, w)$
$T \leftarrow \{\}$
**for all** $i \in V$ **do**
    $Par_T(i) = \arg\max_j w(j, i)$
    $Par_T(i) \leftarrow T \cup \{(Par_T(i), i)\}$
**end for**
**Return:** $T$
```

Algorithm 2 NetInf Algorithm

```
**Require:** Cascades $C = \{(c, \mathbf{t_c})\}$ and #edges $k$
$G \leftarrow \bar{K}$
**for all** $c \in C$ **do**
    $T_c \leftarrow max\_span\_tree(c)$ {Algorithm 1}
**end for**
**while** $|G| < k$ **do**
    **for all** $(j, i) \notin G$ **do**
        $\delta_{j,i} = 0$
        $M_{i,j} \leftarrow \emptyset$
        **for all** $c : t_j < t_i inc$ **do**
            Consider $G \cup \{(j, i)\}$
            **if** $w_c(j, i) \leq w_c(Part_{T_c}(i), i)$ **then**
                $\delta_{j,i} = \delta_{j,i} + w_c(j, i) - w_c(Part_{T_c}(i), i)$
                $M_{j,i} \leftarrow M_{j,i} \cup \{c\}$
            **end if**
        **end for**
    **end for**
    $(j^*, i^*) \leftarrow \arg\max_{(j,i) \in C \setminus G} \delta_{j,i}$
    $G \leftarrow G \cup \{(j^*, i^*)\}$
    **for all** $c \in M_{j^*, i^*}$ **do**
        $Par_{T_c}(i^*) \leftarrow j^*$
    **end for**
**end while**
**Return:** $G$
```

achieve $1 - 1/e \approx 63\%$ of the optimal value for the graph $\hat{G}$ return be the algorithm.The detailed process is shown in algorithm 1 and algorithm 2.

3.2. Influence Estimation

Once the global network structure is learned, we can estimate the influence of a set of source nodes A within a given time window T. Specifically, we want to estimate $\sigma(A, T)$ (Du et al., 2013a),

$$\sigma(A, T) = E[\sum_{i \in V} I\left(t_i \leq T\right)] = \sum_{i \in V} E[I\left(t_i \leq T\right)] \quad (10)$$

$$= \sum_{i \in V} Pr(t_i \leq T) \quad (11)$$

where $I(\cdot)$ is the indicator function and V is the set of all vertices in the graph G. The above problem can be converted to a function of the edge transmission times $\tau_{ji} = t_i - t_j$,

$$\sigma(A, T) = \sum_{i \in V} Pr\{g_i\left(\{\tau_{ji}\}_{(j,i) \in E}\right) \leq T\} \quad (12)$$

where $g_i\left(\{\tau_{ji}\}_{(j,i) \in E}\right) = t_i$ is the shortest path from the source nodes to node i. Given a set of fixed edge transmission times $\{\tau_{ji}\}_{(j,i) \in E}$ and a source node s infected at time 0, the neighborhood $N(s, T)$ of s in a given time window T can be estimated using the Cohen's randomized neighborhood estimation algorithm (Cohen, 1997),

$$|N(s, T) \approx \frac{m - 1}{\sum_{u=1}^{m} r_\star^u} \quad (13)$$

where $r_\star = \min_{i \in N(s,T)} r_i$ is the smallest label within distance T from the source s and will distribute as $r_\star \sim exp(-|N(s, T)|r_\star$, and m is the number of randomized initializations of nodes i over the labelings $r_i \sim exp(-r_i)$. To estimate the neighborhood of a set of A sources, we simply take the union of the neighborhood for each individual source (Du et al., 2013a),

$$N(A, T) = \cup_{s \in A} N(s, T) \quad (14)$$

and the least label $r_\star$ for the neighborhood $N(A, T)$ can be calculated by taking the minimum of the least labels $r_{\star(s)}$ for each of the individual sources s. Similarly, $|N(A, T)|$ can be estimated using Eq. (13) by taking m label samples.

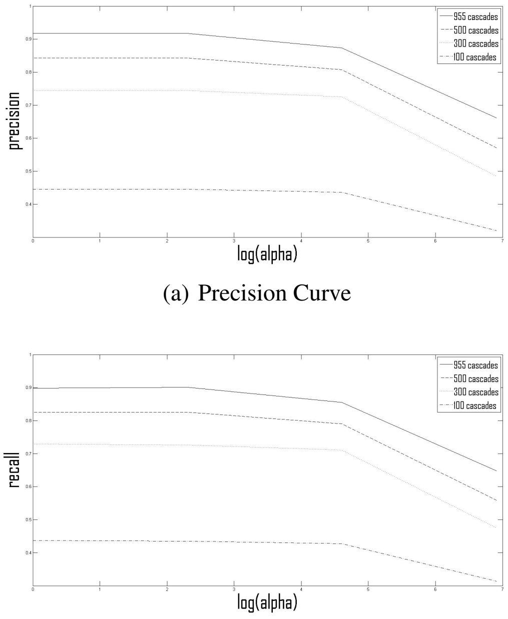
3.3. Influence Optimization

Once we have estimated the influence, our goal is to find out an optimal set of source nodes which maximize or minimize the influence. That is,

$$A*_{(max)} = \arg\max_A |N(A, T)| \quad (15)$$

$$A*_{(min)} = \arg\min_A |N(A, T)| \quad (16)$$

Like in the previous section, $|N(A, T)|$ is a monotonic, non-decreasing function in the source node set A and satisfies the diminishing returns property of submodularity. Therefore, we modify our optimization to include the following constraints,



(a) Precision Curve

(b) Recall Curve

Figure 2. Precision and recall decrease with very large alpha

$$A*_{(max)} = \arg\max_{|A|\leq k} |N(A,T)| \tag{17}$$

$$A*_{(min)} = \arg\min_{|A|\geq k} |N(A,T)| \tag{18}$$

where we are looking for the k most influential and least influential sources, respectively. Eq. (17) can be solved using the same greedy approach for submodular functions as described in the previous section. For Eq. (18), the greedy approach can be used but the optimality guarantee $1 - 1/e$ does not hold. We will give more theoretical details about optimality for the minimization task after running more experiments.

4. Experimental Setup and Results

In this section, we proceed with the experimental evaluation of our proposed methods, as well as a quantitive and qualitative analysis of the corresponding results. Specifically, we organize this section by addresssing the following problems in order: (1) given only timestamp data, learn the most likely network structure, (2) given a network, estimate influence of sources within a given time window, and (3) given only timestamp data, estimate the influence of sources within a given time window.

4.1. Inferring Network Structure

For this experiment, we explore the properties of NETINF algorithm. The data we used is synthetic data generated by tools provided in the SNAP project. Specifically we generate kronecker graph with 1024 nodes and 2048 edges. We tried different number of cascades. To evaluate, we calculate the Precision and Recall of the estimated graph V.S. the ground truth graph. 2(a) and 2(b) shows the Precision and Recall results, respectively. Here alpha is the parameter for exponential distribution, the tranmission distribution in NETINF algorithm. As we can see, when alpha increases, both Precision and Recall decrease, the best alpha is between 0 and 2. Besides, we could observe that as the number of cascades increases, both Precision and Recall increase, this is because with more cascades, we have more information to learn our graph. And with 955 cascades, both Precision and Recall can be above 90%.

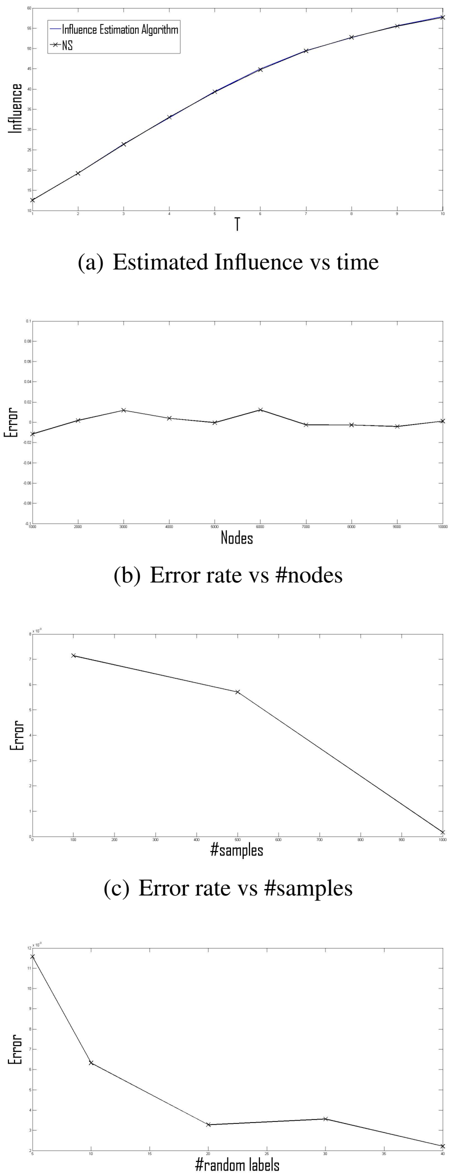
4.2. Estimating influence via known network

For this task, we use a part of the Memetracker dataset (Leskovec et al., 2009) from the Stanford Large Network Dataset Collection (Leskovec, 2014), a collection of news articles and blogs posts from August 2008 to April 2009.

Real-world data. Memetracker dataset tracks quotes and memes that occur most frequently in news media websites and blogs. For each instance of a document (news article or blog post), it contains the URL of the document, time of the post (timestamp), the textual phrases extracted from the document, and hyper-links in the document(links pointing to other documents on the web). It contains 96 million documents and 418 million links, along with 17 million different phrases. About 54% of the total phrases appear in blogs and 46% in news media. It is possible to see how different stories diffuse and fade in the network.

Ground Truth Networks. We preprocess our data to include only the most frequent websites (nodes) based on the number of hyperlinks they create/receive, and consider only the most frequent 343602 textual phrases as contagions. Specifically, we construct 10 ground truth networks ranging from a minimum of 1000 nodes to a maximum of 10000. We wanted to run experiments on larger networks but the computational limitations of our machine did not permit us to do the same. As there are no ground truth networks for the dataset, we use the following method to construct a ground truth network G^* for each node size. Each network G^* has a directed edge between a pair of nodes u and v if a post on site u is linked to a post site v. These networks are subsequently used

Influence Estimation. For this experiment, we only consider the node with the highest out-degree in G^* as the source. To compare the accuracy of our estimated influence



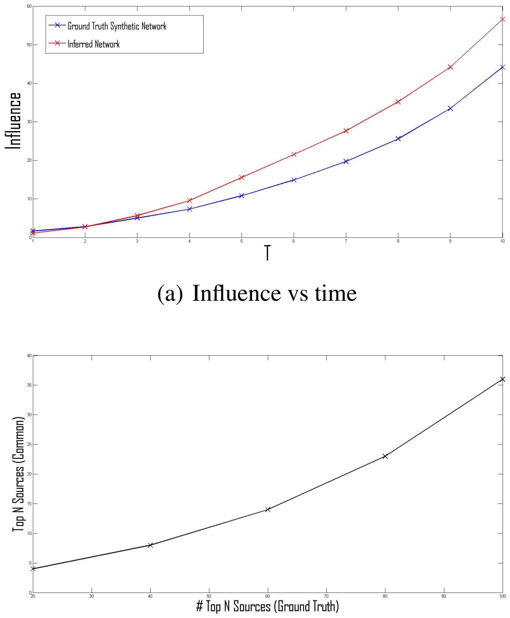
(a) Estimated Influence vs time

(b) Error rate vs #nodes

(c) Error rate vs #samples

(d) Error rate vs #labels

Figure 3. For the MemeTracker dataset (a) estimated influence with increasing time window T, (b) for T = 10 and node size = 4000, error rate with increasing node size, (c) error rate for increasing number of samples, and (d) error rate for increasing number of labels



(a) Influence vs time

(b) Top common sources vs top ground truth

Figure 4. For synthetic network with node size = 1024, (a) network inference - influence estimation integration, and (b) network inference - influence maximization integration

we use the Naive Sampling (NS) approach as the ground truth measure for which we draw 10,000 samples. The heterogenous transmission densities were characterized by a Weibull distribution which is more flexible than the exponential and power-law distributions typically used in such tasks. The scale parameter α, where $\alpha > 0$, and the shape parameter β, where $\beta > 0$, for the Weibull distribution were learned from the data.

Analysis of Results. Figure 3(a) shows the results for the influence estimation algorithm versus the ground truth. It is quite clear, that the algorithm produces the same estimates as the ground truth. Figure 3(b) compares the algorithm with the ground truth for networks with different node sizes (ranging from 1000 to 10000). Again it can be seen, the error rate between the influence estimation algorithm and the ground truth measure is negligible, irrespective of the size of the network. Figures 3(a) and 3(b) were generated using 10000 random samples of sets of waiting times and 5 random labels for each set of waiting times. We wanted to see the impact of the number of random samples and random labels on error rate. As expected, the error rate with respect to ground truth decreases with increasing the number of random samples as well as random labels (Figures 3(c) and 3(d)).

4.3. Estimating influence via unknown network

Most real-world situations such as viral marketing, the spread of infectious diseases, the spread of news and opinions etc. are characterized by only time-stamp data. For example, in virus propagation we usually observe people getting sick without knowing who infected them. Moreover, in such situations it is often important to learn the people (nodes) who are most responsible for propagating the virus. Thefore, it is essential to integrate the task of network inference with influence estimation. For this task, we use the synthetic networks in the form of Kronecker networks (Leskovec et al., 2010) generated by Stanford Large Network Dataset Collection (Leskovec, 2014).

Synthetic Network Generation. We use the core-periphery netowrks (parameter matrix: [0.9 0.5; 0.5 0.3]), which mimic the information diffusion traces in real world networks (Gomez Rodriguez et al., 2010). Similar to the previous task, we use the Weibull distribution to characterize the heterogeneous edge transmission densities. Since, this is a synthetic network there is no real-world data to simulate the probability density functions. Therefore, the parameters α and β were chosen uniformly at random from 0 to 10. The number of nodes and edges in this network were 1024 and 2048, respectively. The synthetic network serves as the ground truth network for this task.

Network Inference-Influence Estimation. The results of the previous task demonstrated the accuracy of the influence estimation algorithm. To show that the network inference task integrates well with influence estimation, it is important to compare the estimated influence across various time windows for the ground truth synthetic network with that of the network inferred from the same ground truth synthetic network. Figure 4(a) shows the plots for the estimated influence versus time window for the ground truth synthetic network and the inferred network from the same ground truth synthetic network. The plots are quite close to each other and, moreover, have a similar shape. This shows that the inferred network is quite close to the ground truth synthetic network, and the network inference problem can be integrated with influence estimation problem as one joint problem.

Network Inference-Influence Maximization. We ran a similar experimental setup to the one above to see if the network inference problem can integrate well with the influence maximization problem. Figure 4(b) shows the top n sources obtained from the ground truth synthetic network on the horizontal axis, and the number of common sources in the top n sources of the inferred network and ground truth synthetic network the vertical axis. As it can been, the network inference-influence maximization integration does not perform as well as the network inference-influence estimation.

5. Conclusion

In this paper we have explored a graph structure learning algorithm and an influence estimation algorithm. Based on that, we have proposed a method to estimate influence without any knowledge about the true graph structure, which is the case in most real world application. Experiments have shown that the graph learned by NETINF provides us a reliable graph structure to estimate the influence when the ground truth graph structure is unknown. Future works include compare our approach with some non-graph based influence estimation method such as SIR(Hethcote, 2000) and SIS(Newman, 2003). We would also want to explore how to set the number of edges in NETINF algorithm automatically.

References

Adar, Eytan, Zhang, Li, Adamic, Lada A, and Lukose, Rajan M. Implicit structure and the dynamics of blogspace. In *Workshop on the weblogging ecosystem*, volume 13, 2004.

AlSumait, Loulwah, Barbará, Daniel, and Domeniconi, Carlotta. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 3–12. IEEE, 2008.

Anderson, Roy M, May, Robert M, and Anderson, B. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

Chen, Wei, Wang, Yajun, and Yang, Siyu. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208. ACM, 2009.

Cohen, Edith. Size-estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences*, 55(3):441–453, 1997.

Du, Nan, Song, Le, Yuan, Ming, and Smola, Alex J. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems*, pp. 2789–2797, 2012.

Du, Nan, Song, Le, Gomez-Rodriguez, Manuel, and Zha, Hongyuan. Scalable influence estimation in continuous-time diffusion networks. In *Advances in Neural Information Processing Systems*, pp. 3147–3155, 2013a.

Du, Nan, Song, Le, Woo, Hyenkyun, and Zha, Hongyuan. Uncover topic-sensitive information diffusion networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pp. 229–237, 2013b.

Galuba, Wojciech, Aberer, Karl, Chakraborty, Dipanjan, Despotovic, Zoran, and Kellerer, Wolfgang. Outtweeting the twitterers-predicting information cascades in microblogs. In *Proceedings of the 3rd conference on Online social networks*, pp. 3–3. USENIX Association, 2010.

Goldenberg, Jacob, Libai, Barak, and Muller, Eitan. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3): 211–223, 2001.

Gomez Rodriguez, Manuel, Leskovec, Jure, and Krause, Andreas. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1019–1028. ACM, 2010.

Goyal, Amit, Bonchi, Francesco, and Lakshmanan, Laks VS. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011.

Granovetter, Mark. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420, 1978.

Guille, Adrien, Hacid, Hakim, Favre, Cécile, and Zighed, Djamel A. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(1):17–28, 2013.

Hethcote, Herbert W. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

Kempe, David, Kleinberg, Jon, and Tardos, Éva. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Khuller, Samir, Moss, Anna, and Naor, Joseph Seffi. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

Kleinberg, Jon. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4): 373–397, 2003.

Lawless, Jerald F. *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons, 2011.

Leskovec, Jure. Stanford large network dataset collection, February 2014. URL `http://memetracker.org/data/index.html`.

Leskovec, Jure, Adamic, Lada A, and Huberman, Bernardo A. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

Leskovec, Jure, Backstrom, Lars, and Kleinberg, Jon. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 497–506. ACM, 2009.

Leskovec, Jure, Chakrabarti, Deepayan, Kleinberg, Jon, Faloutsos, Christos, and Ghahramani, Zoubin. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

Lu, Rong and Yang, Qing. Trend analysis of news topics on twitter.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

Newman, Mark EJ. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

Rodriguez, Manuel Gomez, Balduzzi, David, and Schölkopf, Bernhard. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.

Rodriguez, Manuel Gomez, Leskovec, Jure, and Schoelkopf, Bernhard. Modeling information propagation with survival theory. *arXiv preprint arXiv:1305.3616*, 2013.

Rogers, Everett M. Diffusion of innovations. 1995. *Woerkom van C, Kuiper D, Bos E. Communicatie en innovatie, een inleiding. Alphen aan den Rijn: Samsom*, pp. 2, 1999.

Shamma, David A., Kennedy, Lyndon, and Churchill, Elizabeth F. Peaks and persistence: Modeling the shape of microblog conversations. CSCW '11, pp. 355–358, New York, NY, USA, 2011.

Takahashi, Toshimitsu, Tomioka, Ryota, and Yamanishi, Kenji. Discovering emerging topics in social streams via link anomaly detection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 1230–1235. IEEE, 2011.