SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2018, Vol. 61 092101:1–092101:16
doi: 10.1007/s11432-016-9122-x

Inferring diffusion networks with life stage heterogeneity

Tong ZHAO[1], Guojie SONG[1*] & Xinran HE[2]

[1]*Key Laboratory of Machine Perception (MOE), Peking University, Beijing* 100871*, China;*
[2]*Computer Science Department, University of Southern California, Los Angeles* 90089-0911*, USA*

Received 18 July 2016/Accepted 10 May 2017/Published online 2 November 2017

Abstract A network inference problem focuses on discovering the structure of a diffusion network from observed cascades. This problem is significantly more challenging in several settings in which this type of an inference is desirable or necessary because of heterogeneity in the diffusion process. The heterogeneity of the diffusion process in different life stages results in the inaccuracy of a common assumption of constant influence strength. In this study, a Life Stage Heuristic (LSH) method is proposed to model life stage heterogeneity by decoupling the popularity level of an item under propagation from a true strength of social ties to improve inference accuracy. The proposed LSH is incorporated into almost all existing state-of-the-art network inference algorithms to improve estimation accuracy with only minimal changes in the implementation and maintaining the same running time. Additionally, NetRate, NetInf, and ConNIe are used as three examples to demonstrate the power of the proposed method. Furthermore, clustering of cascades prior to the LSH is proposed to eliminate noise, and the optimized method is termed as Clustered Life Stage Heuristic (CLSH). Extensive experiments on synthetic and real world datasets indicate that both LSH and CLSH methods significantly improve the accuracy of network inference.

Keywords network inference, life stage heterogeneity, social influence, information diffusion, clustering cascade

Citation Zhao T, Song G J, He X R. Inferring diffusion networks with life stage heterogeneity. Sci China Inf Sci, 2018, 61(9): 092101, doi: 10.1007/s11432-016-9122-x

1 Introduction

An understanding of the processes and dynamics of information diffusion through networks plays a fundamental role in a variety of domains including evaluating the effects of networks in marketing [1–3], monitoring the spread of news, opinions and scientific ideas via citation networks [4–6], and detecting the spread of erroneous information [7].

The underlying diffusion network (e.g., networks on who influenced whom) is often hidden in practical applications. For example, although it is only possible to record the time when symptoms of disease are observed in a patient, the channel of infection (such as a close geographical contact with friend who is already infected) is typically confusing. It is easy to obtain the time at which a web site article is published. However, in a few cases, editors do not refer to their information source, and this makes it difficult to obtain the diffusion network of news. For example, individuals swarm into a brand-new restaurant and share remarks on apps after a fancy dinner. Nevertheless, customers rarely mention the

[*] Corresponding author (email: gjsong@pku.edu.cn)

person who introduced them to the restaurant. It is possible to observe the time at which individuals buy or sell a stock although it is difficult to determine the individual who influenced the action. Therefore, several interesting models were developed to automatically infer diffusion networks from observed cascades such as timestamps when users post a blog on certain topics or purchase products [8–18].

Most previous studies on network inference problems assume that the strength of influence between pair of nodes remains constant throughout the entire life of the diffusion process. It is easily observed that this assumption is unrealistic. For example, in the case of the propagation of a story in a microblog, it is more likely that a user is influenced by friends to talk about the story when it is popular. In contrast, it is less likely that an individual can influence his/her friends to be interested in a story when the story becomes stale and less popular. In this study, the violation of the assumption of homogeneity on two real-world cascade datasets is demonstrated by empirically estimating diffusion speed.

An unsatisfactory solution of the network inference problem can result from ignoring the heterogeneity of influence strength in different stages. It is assumed that it is observed that a user v posts the same contents immediately after u. Existing network inference algorithms treat this as strong evidence that a user u exerts strong influence on v. However, it is also possible that v responds fast simply because the story is extremely popular and any post from his/her friends leads to a similarly rapid response. Conversely, if the above scenario occurs when the story is no longer popular, then v's rapid response to u's post can be safely treated as strong evidence that u significantly influences v. The confusion between a strong relationship between friends and the popularity of a story leads to a failure in this scenario.

In this study, the heterogeneity of influence strength in different life stages of diffusion process is incorporated to improve the accuracy of network inference. A heuristic referred to as the Life Stage Heuristics (LSH) is designed. Extant studies [9, 10, 13] typically assumed that the influence strength between a pair of nodes is only associated with the strength of social ties, and this is time-invariant. The proposed LSH uses the apparent influence strength instead of this, and it is determined by both social tie strength and popularity of information. The apparent influence strength in the diffusion process and the underlying strength of social ties are differentiated between individuals. It is assumed that the apparent influence strength models the actual diffusion speed of the information or the activation probability observed in the cascades. This varies in different stages of the diffusion process with changes in the popularity of the information. Conversely, the underlying strength of social ties between individuals remains stable throughout the diffusion process. The proposed LSH focuses on discovering the underlying strength of social ties as opposed to the apparent influence strength because the former is considered a more accurate indicator for the strength of a relationship and the inference of network structure. In more precise terms, the apparent influence strength is decoupled into a product of the strength of a social tie and the popularity level of the information. This formulation decomposes the true strength of social ties from the popularity of the information as a confounding factor.

The proposed LSH is incorporated into almost all existing network inference algorithms by substituting the influence strength with popularity sensitive apparent influence strength. The simplicity of the multiplicative form makes it extremely easy to apply LSH to existing algorithms with only minimal changes in implementation and no increase in running time. In this study, three state-of-the-art network inference algorithms NetInf [12], ConNIe [19], and NetRate [11] are used as examples to demonstrate the manner in which the LSH is incorporated to significantly improve inference accuracy of both synthetic and real-world datasets.

2 Preliminaries and related work

This section initially establishes the preliminaries of network inference problem and the Continuous-time Independent Cascade (CIC) model used in the study. Additionally, several extant studies on network inference that either consider or do not consider heterogeneity are also reviewed.

2.1 Network inference problem

Generally, a diffusion network is modeled as a matrix $A = \{\alpha_{u,v}\}$, where $\alpha_{u,v}$ corresponds to a parameter that represents the strength of influence that a user u exerts on v. Based on the diffusion model, $\alpha_{u,v}$ denotes the parameters of the delay distribution for information to propagate from u to v (for example in NetRate [11]) or the probability that v publishes a related post given that u has previously published a post (for example in ConNIe [19]). The network inference problem is formally defined as accurately discovering all parameters $\alpha_{u,v}$ from the observed cascades $C = \{c_1, \ldots, c_C\}$. Each cascade c_i provides a trace of propagation of item i among the nodes of the network, i.e., $c_i = (t_1^i, t_2^i, \ldots, t_m^i)$ as a node v_m adopts item i at time t_m^i. A node is considered as uninfected if its activation time follows the observation window T.

2.2 Diffusion model

The network inference problem was studied under various diffusion models. Popular models include cascade models (for example, NetInf [12], ConNIe [19], and NetRate [11]) and point process models (such as MMHP [14] and LowRankSparse [15]).

This study focuses on a popular cascade model, namely the CIC model. The approach is easily applied to other diffusion models as well. The CIC model is widely adopted in network inference literature [11, 12, 19]. The most general form of the CIC model is as follows. Each edge (u, v) is associated with both a delay distribution $p_{u,v}(t)$ and an activation probability $ap_{u,v}$. When node u is newly activated at time t_u, for every neighbor v that is still inactive, a biased coin with success probability $ap_{u,v}$ is flipped to determine whether or not v is activated. If the attempt succeeds, then a delay time Δt is further drawn from the distribution $p_{u,v}(t)$ and v becomes active at time $t_u + \Delta t$. If multiple nodes succeed in activating the same inactive node, then the activation time is considered as corresponding to the earliest time.

2.3 Heterogeneous influence

This is not the first study to consider heterogeneity in diffusion processes. Various extant studies considered the heterogeneity of influence in social networks to improve the accuracy of network inference [9, 10, 13]. For example, the Topic Cascade algorithm [9] focuses on inferring a diffusion network from text-based cascades in which the diffusion rate depends on the similarity of the contents. The Kernel Cascade algorithm [10] assumes that the delay distributions are different for different edges in a diffusion network and that the delay distributions are not limited to traditional distribution models (e.g., Exponential model, Power-law model, and Rayleigh model), which are depicted with a series of kernel functions. Additionally, Wang et al. [13] proposed an innovative MMRate algorithm to perform multi-aspect multi-pattern network inference that assumed that the social tie of a pair of nodes changes on different message aspects, and therefore corresponds to a multi-aspect. Furthermore, the diffusion speed and scale are different in each cascade, and this corresponds to a multi-pattern. MMRate proposed using an Expectation Maximization algorithm to estimate the parameters of social ties and cascade patterns.

The present study is orthogonal to the fore-mentioned studies. Previous studies consider the heterogeneity on different edges in the network while continuing to assume constant strength of influence throughout the whole diffusion process. Conversely, the present study considers heterogeneity of influence in the time dimension. As a result, the proposed LSH is easily incorporated with the above algorithms.

The heterogeneity considered in the proposed model is that social influence is determined by both the strength of social tie between nodes and the current popularity level of message that is time variant. A few previous studies focus on learning the pattern of popularity. Specifically, K-SC [20] clustering method extracts time series of popularity and aims to distinguish several distinct shapes of time series to reflect the underlying popularity pattern. They propose a similarity metric for clustering that is invariant to scaling and shifting and determine that the patterns of popularity are mainly divided into six groups. Furthermore, SPIKEM [21] is an analytical method that models the main factors affecting popularity and predicts popularity value evolution. The LSH model in the present study initially incorporates a network inference problem with the heterogeneity induced by popularity to further improve inference accuracy.

3 Popularity level heterogeneity estimation

In this section, an empirical analysis on two real world cascades datasets is performed to demonstrate that the assumption of constant influence strength is not applicable. Subsequently, a method is detailed to empirically estimate the popularity of information under propagation from the observed cascades. The estimated popularity level is used in the LSH to model the apparent influence strength.

3.1 Existence of influence strength heterogeneity

In several network inference models [11], the influence strength parameter $\alpha_{u,v}$ models the speed of diffusion, namely the time taken for the information to propagate from user u to user v. The average diffusion speed is empirically estimated as the inverse of delay time in two real world cascade datasets, namely the Sina microblog and US Patent, to demonstrate the heterogeneity of influence strength.

In the Sina microblog, each node corresponds to a single user, and the propagation of stories consists of cascades. The diffusion pathway is given by the hyperlinks of user v responding to user u's posts. Conversely, each node represents an inventor in the US Patent dataset. Each cascade is extracted as a diffusion tree in which an innovative idea propagates among inventors following the citing relationship. The diffusion pathway is given by the citation relationship between inventors in the dataset.

The estimation process works as follows: the span of each cascade is initially normalized to $[0, 1]$, and the interval is split evenly into ten bins. Each bin is treated as one stage of the diffusion process. The average diffusion speed in each bin is estimated as the averaged inverse of delay time $\frac{1}{\Delta_t}$ by using the available explicit diffusion pathway information in the datasets. Specifically, Δ_t is computed in a normalized time scale. The plot of two typical cascades from each dataset is shown in Figure 1.

The results clearly demonstrate the deviation based on the assumption of constant diffusion speed. Another interesting observation relates to the manner in which the diffusion speed depending on the diffusion stages varies significantly over different cascades and datasets. With respect to the microblog datasets, the diffusion speed typically decreases when the story under propagation becomes stale and less interesting as time elapses. In contrast, with respect to the Patent dataset, a newly invented product gradually gains popularity leading to increases in the diffusion speed.

3.2 Approximation of popularity by fraction of activated nodes

The popularity level is not directly observable. Additionally, approximation by diffusion speed requires diffusion pathway information, and this is not usually available in real-world datasets. As a result, it is necessary to obtain an approximation for the popularity level in different stages in the diffusion process. In the study, it is proposed that the fraction of activated nodes in each stage of the diffusion process is used as a surrogate for measuring the popularity level.

It is assumed that $I^c(t)$ denotes the number of activated nodes in cascade c in a normalized time interval $[0, t] \subseteq [0, 1]$, and that $I^c(1)$ refers to the total number of activated nodes in the cascade. The popularity level is approximated by using the following formula:

$$P^c(t) = \left(I^c\left(t + \frac{L}{2}\right) - I^c\left(t - \frac{L}{2}\right)\right)/I^c(1).$$

Hence, the fraction of activated nodes in the windows of length L, $[t - \frac{L}{2}, t + \frac{L}{2}]$ are used as the estimated popularity level at time t.

Experiments are performed to demonstrate that the proposed surrogate popularity level corresponds to a valid approximation. The top 80 and 50 cascades in the Patent and Sina microblog datasets, respectively, with the largest number of activated nodes are selected. With respect to each cascade, the time span is initially normalized to $[0, 1]$, and it is split into 10 bins. The true popularity level is as

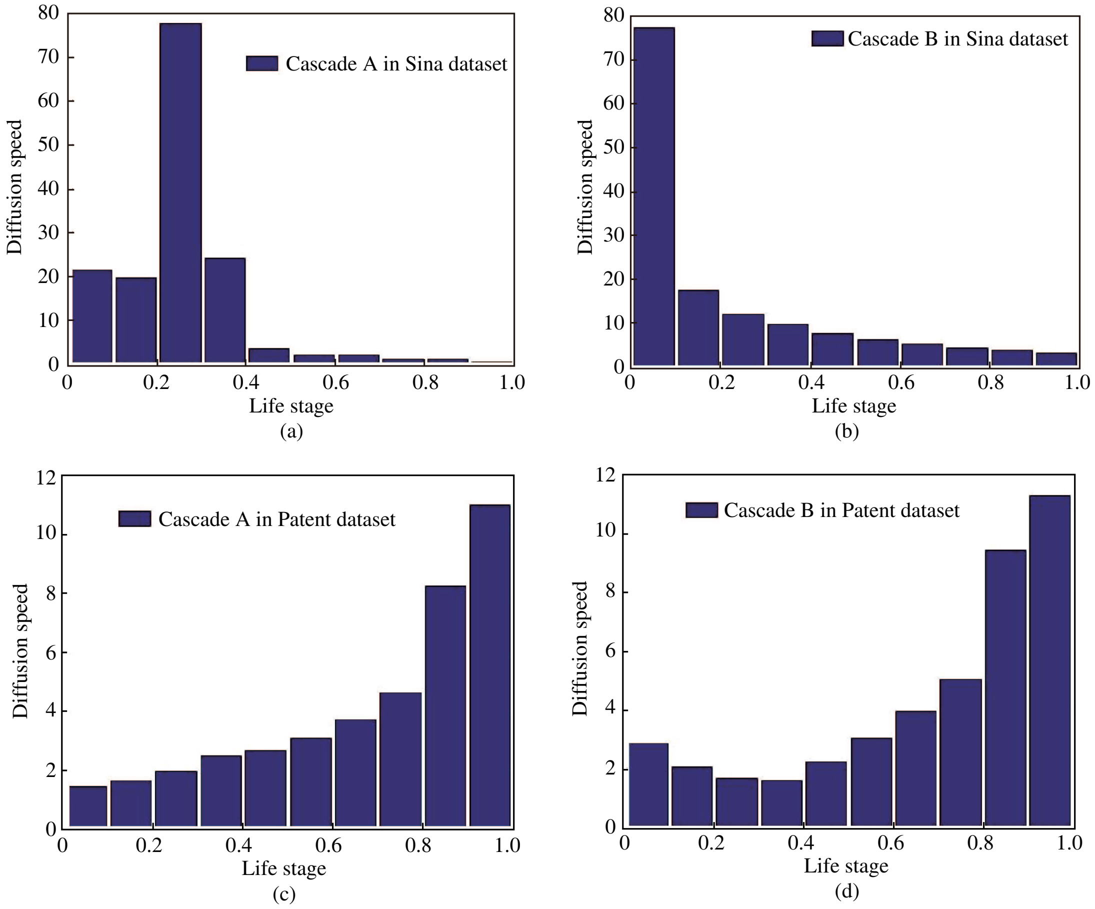


Figure 1 (Color online) The diffusion speed in different stages of diffusion processes in the Sina microblog dataset and US Patent dataset. (a) Cascade A in Sina dataset; (b) cascade B in Sina dataset; (c) cascade A in Patent dataset; (d) cascade B in Patent dataset.

Table 1 Correlation coefficients between ground truth diffusion speed and the number of activated nodes in a time unit

Coefficient	Strength	Percentage in Patent	Percentage in Sina
0–0.19	Very weak	0.050	0.02
0.20–0.39	Weak	0.125	0.02
0.40–0.59	Moderate	0.175	0.06
0.60–0.79	Strong	0.300	0.08
0.80–1	Very strong	0.350	0.82

assumed constant in each stage and is estimated as the average diffusion speed with the method described in Subsection 3.1. The true popularity level and the surrogate popularity at each activation time t_u^c are compared by computing the Pearson correlation for each cascade c. The distribution of the correlation coefficient in the two datasets is shown in Table 1. As shown in the table, more than 60% cascades and 80% cascades in the Patent and Sina microblog datasets, respectively, exhibit a strong correlation ($r \geqslant 0.6$) with the true popularity level. The strength of correlation is defined based on [22].

In addition to approximating the popularity level, the following methods are proposed to further improve the accuracy of the estimation:

(1) Dealing with root node. Each cascade has a starting node whose activation time always corresponds to 0 after normalization. The inclusion of the starting node incorrectly increases popularity level at the beginning of the diffusion process. As a result, the starting node is removed to ensure that the estimation is unbiased.

(2) Clustering cascades. The popularity level that is estimated for each individual cascade could be noisy due to the limited number of activated nodes or an anomaly in the activation. In order to achieve a more accurate estimation, it is proposed that the cascades should be initially clustered to groups. Subsequently, the averaged popularity level with respect to the cascades in each cluster is used as the estimation for each individual cascade inside the cluster.

Therefore, the individual popularity level for each cascade in the method described above is initially computed. This is followed by performing hierarchical clustering to separate the cascades to different groups. The cascades in the same group use an averaged popularity level as opposed to their own individual estimation. In this case, the proposed method is termed as Clustered Life Stage Heuristic (CLSH).

4 Proposed algorithm

This section describes the manner in which the proposed LSH is applied to state-of-the-art network inference algorithms to incorporate the popularity level heterogeneity. Three commonly used algorithms, namely NetRate, ConNIe, and NetInf are used as examples. The simple multiplicative form of LSH makes it extremely applicable with only minimal changes and without any increases in the running time of the algorithms.

4.1 Life stage heuristics

In the proposed LSH method, the apparent influence strength is distinguished from the the strength of social ties between nodes. The former explains the observed activation pattern while the latter represents the true strength of social ties that should be inferred from the cascades. The Apparent Influence Strength is modeled as a function of a life stage, namely $\hat{\alpha}^c_{u,v}(t), t \in [0,1]$ as the strength of influence that u exerts on v at life stage t in cascade c. Conversely, it is assumed that the strength of social ties remains constant throughout the diffusion process, and this is termed as $\alpha_{u,v}$, namely the strength of the relationship between u and v.

Existing algorithms use constant social tie strengths $\alpha_{u,v}$ to model the observed cascades. In the proposed LSH, Apparent Influence Strength $\hat{\alpha}^c_{u,v}(t)$, is used to incorporate the heterogeneity in different life stages. The $\hat{\alpha}_{u,v}(t)$ is modeled as the product of social tie strength $\alpha_{u,v}$ and the popularity level $P^c(t)$ as follows:

$$\hat{\alpha}^c_{u,v}(t) = P^c(t) \cdot \alpha_{u,v}.$$

As a result, the application of the proposed LSH is extremely simple. The influence strength parameter $\alpha_{u,v}$ is simply replaced by $\hat{\alpha}_{u,v}(t)$. The form of $\hat{\alpha}_{u,v}(t)$ is very simple as a product of original parameter and the popularity level that is estimated in advance. This guarantees that the desired properties of the algorithm are preserved after the substitution, for example convexity in NetRate, ConNIe, and submodularity in NetInf.

4.2 Incorporation with existing algorithms

In more concrete terms, the three algorithms are briefly introduced initially, and then the details of the manner in which the proposed LSH is applied to incorporate the proposed heterogeneity are then described.

LSH-NetRate. NetRate algorithm assumes that the cascades are generated from the CIC model where the activation probability $ap_{u,v}$ corresponds to 1 for all edges. The algorithm solves the network inference problem by estimating the parameter $\alpha_{u,v}$ of the delay distribution associated with each edge (u, v). Thus, NetRate performs maximum likelihood estimation by solving a convex optimization problem with $\alpha_{u,v}$ as variables. Additionally, $\alpha_{u,v}$ is substituted with $\hat{\alpha}^c_{u,v}(t_v)$ to preserve the convexity of the problem. Hence, the new transmission likelihood function that represents the probability that node j succeeds in infecting node i by following an exponential distribution is as follows (e is Natural Constant):

$$\hat{f}^c(\Delta t_{u,v}; \hat{\alpha}^c_{u,v}(t_v)) = \hat{\alpha}^c_{u,v}(t_v) \cdot \mathrm{e}^{-\hat{\alpha}^c_{u,v}(t_v)\Delta t_{u,v}} = P^c(t_v) \cdot \alpha_{u,v} \cdot \mathrm{e}^{-P^c(t_v)\cdot\alpha_{u,v}\Delta t_{u,v}}.$$

The survival function corresponds to the likelihood that node v is not infected by node u until time t_v by considering the current popularity level of the message, which is as follows:

$$\hat{S}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v)) = 1 - \int_0^{t_v - t_u} \hat{f}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v))\mathrm{d}\Delta t = \mathrm{e}^{-P^c(t_v)\cdot(t_v-t_u)\alpha_{u,v}}.$$

The hazard function corresponds to the probability that node v is not infected by node u until time t_v and that the infection occurs within a short interval after t_v, and this as follows:

$$\hat{H}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v)) = \frac{\hat{f}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v))}{\hat{S}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v))} = P^c(t_v)\cdot\alpha_{u,v}.$$

The target function of NetRate algorithm models the likelihood that each infected node v in each cascade fails to activate the nodes whose recorded time occurs after T and that v is activated by one of the previously infected nodes, namely the probability that the observed cascade set occurs. The new target function considers the heterogeneity of influence strength in different life stages as follows:

$$\begin{aligned} L(C;A) &= \log \prod_{t^c\in C}\prod_{t_v\leqslant T}\prod_{t_m>T}\hat{S}^c(\Delta t_{v,m};\hat{\alpha}^c_{v,m}(T))\prod_{k:t_k<t_v}\hat{S}^c(\Delta t_{k,v};\hat{\alpha}^c_{k,v}(t_v))\sum_{u:t_u<t_v}\hat{H}^c(\Delta t_{u,v};\hat{\alpha}^c_{u,v}(t_v)) \\ &= \sum_{t^c\in C}\sum_{t_v\leqslant T}\sum_{t_m>T}(-P^c(T)(T-t_v)\alpha_{v,m})\sum_{k:t_k<t_v}(-P^c(t_v)(t_v-t_k)\alpha_{k,v})\cdot\log\left(\sum_{u:t_u<t_v}P^c(t_v)\alpha_{u,v}\right). \end{aligned} \tag{1}$$

The value of each $P^c(t)$ is calculated prior to the MLE process, and thus the convexity of target function is preserved. The new problem is solved in the same manner as its original formulation, and this states that A corresponds to a parameter matrix. The value of each required popularity level $P^c(t)$ and transmission delay are initially computed, the target function is formulated as shown in (1), and the cvx package (an excellent convex optimization program) in a MATLAB environment is then used to solve for the same. Each non-zero item $\alpha_{u,v}$ in cvx output sparse matrix A represents that an edge exists between node u and v and that the inferred strength of their social tie corresponds to $\alpha_{u,v}$.

LSH-ConNIe. The ConNIe algorithm is different from the NetRate algorithm since it assumes that all the edges include the same delay distribution in which the parameter is known. Conversely, the parameters $\alpha_{u,v}$ to be estimated in ConNIe algorithm correspond to the activation probability associated with each edge (u,v). In a manner similar to NetRate, the estimation problem is solved via convex programming. The proposed LSH is incorporated in a similar manner. The $\alpha_{u,v}$ is substituted by $\hat{\alpha}^c_{u,v}(t_v)$. Given that $\hat{\alpha}^c_{u,v}(t_v)$ denotes the probability in ConNIe, additional constraints are included to ensure that $\hat{\alpha}^c_{u,v}(t)$ lies in $[0,1]$. Therefore, the target function in LSH-ConNIe for each node v is as follows:

$$\hat{L}_v(A_{:,v};C) = \prod_{c\in C;t^c_v<\infty}\left(1-\prod_{u;t_u\leqslant t_v}(1-w(t^c_v-t^c_u)\hat{\alpha}^c_{u,v}(t_v))\right)\prod_{c\in C;t^c_v=\infty}\left(\prod_{u\in c;t^c_u<\infty}(1-\hat{\alpha}^c_{u,v}(t_v))\right). \tag{2}$$

Thus, $w(t^c_v - t^c_u)$ corresponds to the delay distribution model that indicates the probability of infection when transmission time delay corresponds to $t^c_v - t^c_u$. The solution to $\min_{A_{:,v}} -\log(\hat{L}_v(A_{:,v};C))$ models the strength of v's incoming edges, which are composed of two parts. The first part is the probability of infected node to be activated by at least one previously infected parent. The second part is the probability for non-infected node that no one succeeds in activating it. Given that $P^c(t)$ corresponds to the real value, the modification preserves the properties of the target function. As noted in ConNIe [19], it is difficult to solve the current form of (2) because its Hessian matrix is indefinite. A change in the variables is required to transform the target function into the MLE problem, and this is possible for highly optimized convex programming tools. This is expressed as follows:

$$\min_{\gamma'_c,B'(:,v)}\sum_{c\in C;t^c_v<\infty}-\gamma'_c-\sum_{c\in C;t^c_v=\infty}\sum_{u\in c;t^c_u<\infty}B'_{uv}+\rho\sum_u \mathrm{e}^{-B'_{uv}}$$

$$\begin{aligned}\text{s.t.}\quad & B'_{uv} \leqslant 0\ \forall\ u, \quad \gamma'_c \leqslant 0\ \forall\ c,\\ & \log\left(e^{\gamma'_c} + \prod_{u;t_u\leqslant t_v}(1+w(t_v^c - t_u^c)(e^{B'_{uv}}-1)P^c(t_v))\right) \leqslant 0,\ \forall\ c,\end{aligned} \tag{3}$$

where $B'_{uv} = \log(1-\alpha_{u,v})$, $\gamma'_c = \log(1-\prod_{u;t_u\leqslant t_v}(1-w(t_v^c-t_u^c)\hat{\alpha}_{u,v}^c(t_v)))$ and ρ denote the sparsity parameter. The problem can be solved by calling SNOPT7 library (an useful convex optimization tool) that returns the estimated parameter matrix B'. The value of $\alpha_{u,v} = 1 - e^{B'_{uv}}$ is computed to obtain the inferred strength of the social tie between node u and v.

LSH-NetInf. NetInf algorithm also assumes that all the edges include the same known delay distribution and focuses on estimating the activation probability. However, NetInf algorithm assumes that the probability only corresponds to value 0 or a constant p. Non-zero activation probabilities correspond to edges in the diffusion network. The algorithm infers the edges via a submodular function maximization. The algorithm greedily adds an edge to the diffusion network with a maximum marginal increment in the likelihood. In the NetInf algorithm, the likelihood that a node v is activated by node u with a delay time Δt is modeled as $ap_{u,v}\cdot p_{u,v}(\Delta t)$, namely the product of the activation probability and the probability density function of the delay distribution. In the proposed LSH, an additional term $P^c(t_v)$, namely $ap_{u,v}\cdot p_{u,v}(\Delta t)\cdot P^c(t_v)$, is included to capture the heterogeneous popularity level when v is activated. Thus, the new defined transmission probability $\hat{P}_c(u,v)$ that considering the message popularity level is as follows:

$$\hat{P}_c(u,v) = \begin{cases} \beta P^c(t_v)w(t_v-t_u), & (u,v) \text{ is network edge},\\ \epsilon P^c(t_v)w(t_v-t_u), & (u,v) \text{ is } \epsilon \text{ edge},\\ 1-\beta, & v \text{ is not infected}, \qquad \text{network edge},\\ 1-\epsilon, & v \text{ is not infected}, \qquad \epsilon \text{ edge},\\ 0, & \text{else}, \qquad t_u \geqslant t_v. \end{cases} \tag{4}$$

ϵ edge represents the external influence in which the activation probability corresponds to ϵ. Additionally, the activation probability of network edge (starting point and end point belonging to network node set) corresponds to β.

Given a particular transmission tree $T(V_T, E_T)$, the probability of a cascade spreading in T is defined as $\hat{P}(c|T)$, and this necessitates that the edges in T should be spread successfully and that those not in T fail to diffuse. This is expressed as follows:

$$\hat{P}(c|T) = \beta^q\epsilon^{q'}(1-\beta)^s(1-\epsilon)^{s'}\prod_{(u,v)\in E_T} P^c(t_v)w(t_v-t_u).$$

Specifically, q corresponds to network edge number, q' corresponds to ϵ edge number in T, s corresponds to the number of network edges that fail to diffuse, and s' corresponds to the number of ϵ edges that fail to diffuse.

The aim involves determining a graph G that maximizes the following expression:

$$\hat{P}(C|G) = \prod_{c\in C}\sum_T \hat{P}(c|T),$$

where $\hat{P}(C|G)$ models the probability that one of the possible propagation tree occurs in each cascade. As noted in NetInf [12], an approximation is used to substitute the sum of probability of all possible propagation trees with only the most likely propagation tree T, and this is as follows:

$$\hat{P}(C|G) = \prod_{c\in C}\sum_T \hat{P}(c|T) \approx \max_T \hat{P}(c|T).$$

In order to determine the most likely cascade propagation tree T for each cascade c, T is initialized as an empty tree. Subsequently, with respect to each node v, the improvement in log-likelihood $\psi(u,v) = \log\hat{P}_c(u,v) - \log(\epsilon P^c(t_y)w(t_y-t_x))$ is calculated for the case in which every other node u of edge (u,v)

is in the most likely propagation tree T. The parent of node v in T corresponds to a u that results in the largest $\psi(u, v)$. This process is repeated to build the tree T of cascade c.

The submodularity of the target function is proven by the original method. The modification does not change the properties of the target function. The Greedy algorithm is used to efficiently solve this problem.

The greedy algorithm begins by initializing graph G as an empty graph and generates the most likely propagation tree T_c for each cascade c. Users are asked to provide the expected number of edges k in the inferred graph. When the edge number of the current G is less than k, then the edge that results in the largest marginal gain $\delta_{u,v}$ is selected and added into G as follows:

$$\delta_{u,v} = \sum_{c:t_u<t_v \ \& \ \psi_c(u,v)>\psi_c(Par_{T_c}(v),v)} \psi_c(u,v) - \psi_c(\mathrm{Par}_{T_c}(v),v). \tag{5}$$

It should be noted that each $\psi(m, n)$ in (5) is computed based on the graph of $G \cup \{(u, v)\}$. This process is repeated, and one edge per time is added until $|G| = k$, and thus the inferred network graph G is obtained.

5 Experiments

In this section, the proposed LSH is incorporated to three state-of-the-art network inference algorithms, namely NetRate, ConNIe, and NetInf. The method is initially tested on synthetic datasets. The true diffusion network and popularity heterogeneity are controlled, and thus this set of experiments serves as proof of Concept. The experiments are used to examine the manner in which the parameters in LSH influence the performance. Subsequently, an evaluation of LSH on the US Patent dataset and Meme-Tracker dataset are provided to demonstrate that the proposed LSH achieves a significant improvement when compared to existing methods.

In all the experiments, the area under the precision recall curve and the F1 score and the likelihood of held-out cascades are used as measures in which precision corresponds to the fraction of edges in the inferred network present in the true network. Additionally, recall corresponds to the fraction of edges of the true network present in the inferred network. The F1 score is defined as follows:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

5.1 Experiments on synthetic dataset

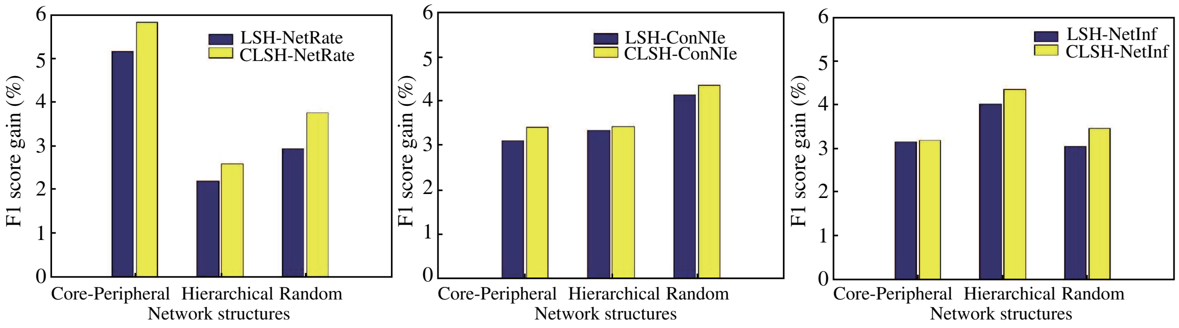
5.1.1 *Data generation*

Synthetic networks are generated with 256 nodes. The generated networks have Random, Hierarchical Community, and Core Peripheral structures that follow the Kronecker Graph model [23] with parameter matrices $[0.5, 0.5; 0.5, 0.5]$, $[0.9, 0.1; 0.1, 0.9]$, and $[0.9, 0.5; 0.5, 0.3]$, respectively. The cascades are generated by following the CIC model. The activation probability for all edges is set to 1, and the exponential distribution is used as the delay distribution. The strength of the social tie is represented as the delay distribution parameter and are sampled uniformly from $[0, 1]$. A node is chosen uniformly at random as the start node. The time window is set as 1 in the experiments.

The life stage popularity heterogeneity is set as a piece-wise constant function. The interval $[0, 1]$ is evenly split into five stages, and the popularity levels are set by following empirical observations. Thus, the bins of the four typical empirical cascades in Figure 1 are initially coarsened from 10 to 5. Subsequently, prior to generating each cascade, one of the four typical heterogeneity patterns is randomly selected after coarsening as shown in Table 2. The window length L in LSH is set to 0.1, and the number of clusters in CLSH is set to 3 in the synthetic experiments if not mentioned otherwise.

Table 2   Popularity patterns in synthetic data

Id	P^c function
1	[0.2695; 0.6742; 0.0319; 0.0183; 0.0060]
2	[0.6441; 0.1494; 0.0942; 0.0648; 0.0474]
3	[0.0746; 0.1091; 0.1408; 0.2045; 0.4710]
4	[0.1122; 0.0735; 0.1206; 0.2080; 0.4857]



(a) (b) (c)

Figure 2   (Color online) Performance of (C)LSH-NetRate (a), (C)LSH-ConNIe (b), (C)LSH-NetInf (c) with respect to various network structures. The x-axis denotes different synthetic network structures.

5.1.2   *Results*

The proposed LSH and the CLSH (referred as (C)LSH) are incorporated to all three network inference algorithms, namely NetInf, NetRate, and ConNIe on all three synthetic networks with different structures. With respect to each network, 500 synthetic cascades are generated. Figure 2 plots the relative gain in max F1 score for (C)LSH methods when compared to the original algorithms. The relative gain in max F1 score is defined as follows:

$$(F_{1\text{(C)LSH-origin}} - F_{1\text{origin}})/F_{1\text{origin}},$$

where the original can correspond to either NetInf, NetRate, or ConNIe. The gain on AUC is similar to that of F1 score and is omitted.

The results indicate that the proposed LSH achieves a consistent improvement for all network inference algorithms and across all network structures. On an average, the proposed methods obtain 4% relative improvement with respect to the F1 score. The most significant advantage of the proposed method is that the gain is almost obtained for free without major changes to existing method and increases in the running time.

Several factors significantly influence the performance of a network inference algorithm. This is followed by examining the manner in which the performance of algorithms depends on the number of cascades and most interestingly the manner in which the algorithms perform when the degree of heterogeneity increases.

5.1.3   *Performance vs. cascade number*

Experiments related to (C)LSH algorithms are initially performed on a Kronecker network with a Core-Peripheral structure to examine the effect of number of cascades. As shown in Figure 3, the application of the proposed methods achieves significant improvement when compared to the original algorithms on both max F1 score and AUC (Area Under Curve) in any number of cascades. The advantage of the proposed method is more significant when the number of cascades is limited. The proposed LSH succeeds in accurately inferring a network with a limited number of cascades since decoupling the popularity level from the true strength of social ties leads to less noisy observations.

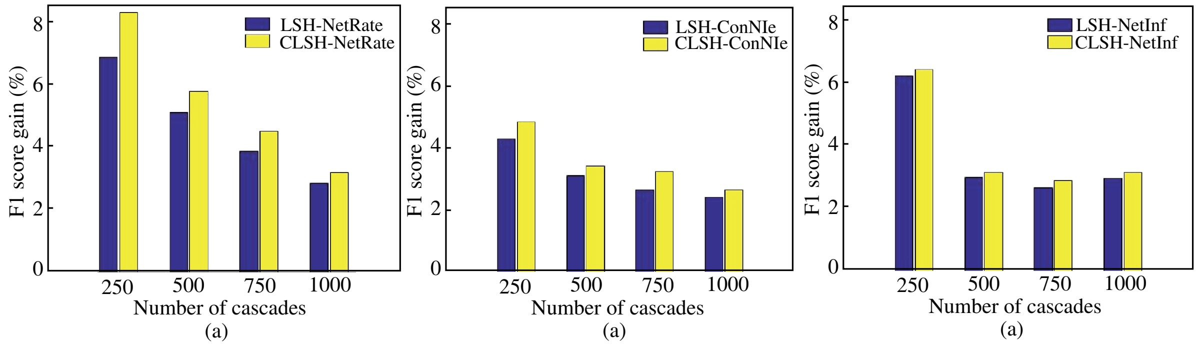


Figure 3 (Color online) Performance of (C)LSH-NetRate (a), (C)LSH-ConNIe (b), (C)LSH-NetInf (c) with respect to different numbers of cascades. The x-axis denotes a different number of cascades.

Table 3 Popularity patterns with respect to different standard deviations

Id	P^c function	Level
1	[0.2; 0.2; 0.2; 0.2; 0.2]	0
2	[0.2; 0.4; 0.3; 0.05; 0.05]	0.1541
3	[0.1; 0.5998; 0.3; 0.0001; 0.0001]	0.2549
4	[0.03; 0.8998; 0.07; 0.0001; 0.0001]	0.3924

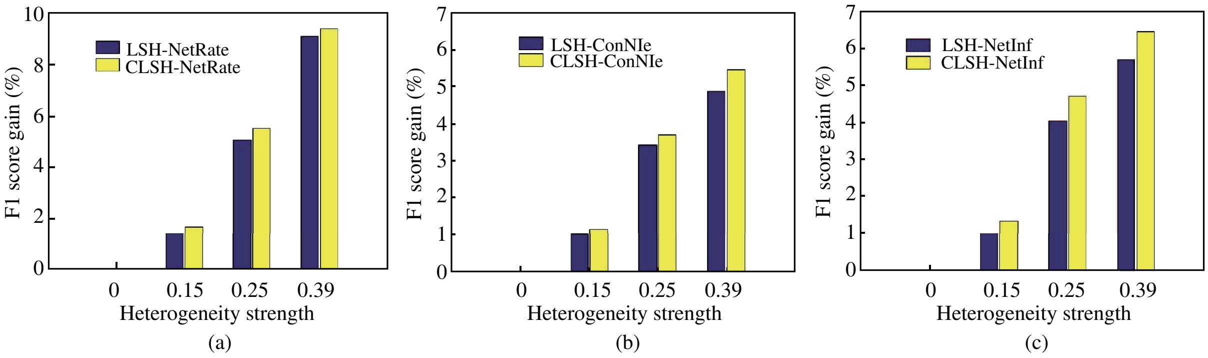


Figure 4 (Color online) Performance of (C)LSH-NetRate (a), (C)LSH-ConNIe (b), (C)LSH-NetInf (c) with respect to different heterogeneity levels. The x-axis denotes different levels of heterogeneity.

5.1.4 *Performance vs. heterogeneity level*

This is followed by examining the performance of the proposed methods under different heterogeneity levels. The standard deviation of $P^c(t)$ is used to measure the level of heterogeneity. Thus, the popularity level with similar shape but different scales are selected as in Table 3. Experiments in Figure 4 show that the proposed method leads to larger gain when compared to the original algorithms with respect to increases in the level of heterogeneity. It should be noted that the methods are exactly the same as the original algorithm when there is no heterogeneity. This corresponds to a zero gain when the popularity level corresponds to a constant function.

5.1.5 *Likelihood of held-out cascades*

In addition to a more accurate inference of the diffusion network structure, the results demonstrate that the proposed LSH also leads to better diffusion modeling by comparing the likelihood of held-out cascades. The experiment involves initially executing baseline algorithms and the proposed methods to infer the diffusion network. Subsequently, the likelihood of the held-out cascades are computed again based on both the original CIC model and the proposed LSH with the inferred network structure. The results are shown in Figure 5. The proposed LSH is incorporated into the CIC model, and this leads to improvements approximately corresponding to 8% and 1.2% when compared to those of the CIC model

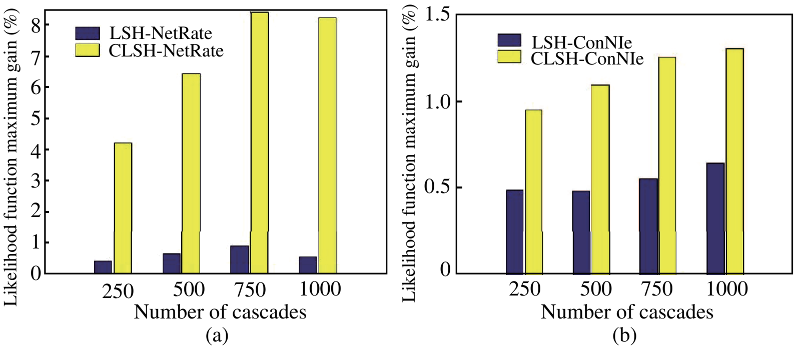


Figure 5 (Color online) Performance on the likelihood function maximum. (a) (C)LSH-NetRate; (b) (C)LSH-ConNIe.

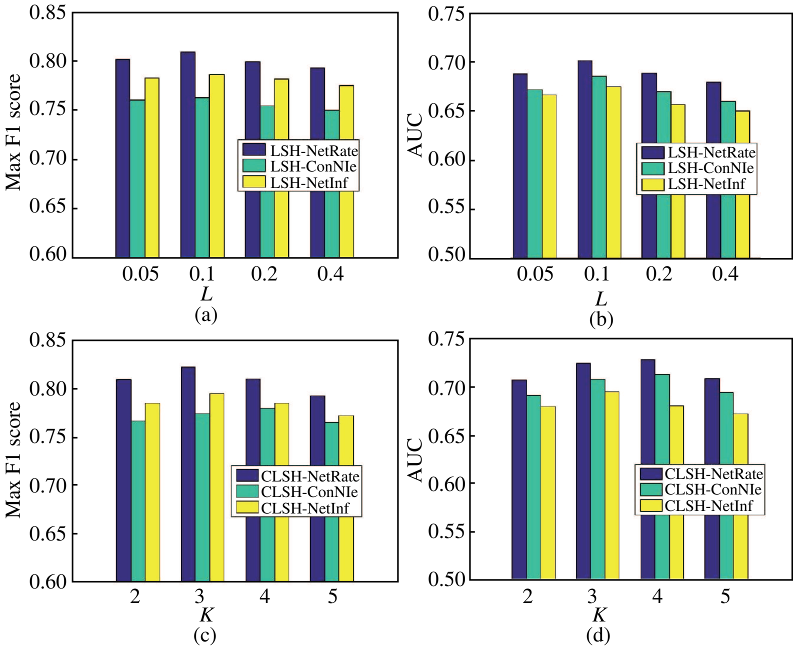


Figure 6 (Color online) Performance with respect to different parameter values L and K. (a) Max F1 score on various L; (b) AUC on various L; (c) max F1 score on various K; (d) AUC on various K.

with parameters inferred by NetRate and ConNIe, respectively. The observations further confirm the heterogeneity in the diffusion process and validate that the proposed LSH fits with real-world diffusion processes better than the traditional CIC model without life stage heterogeneity.

5.1.6 *Setting of window length L*

Another important parameter that should be examined corresponds to the length of windows L in estimating the popularity level. The experiment involves different settings of L as shown in Figure 6. As shown in the figure, the proposed LSH method is not sensitive to the setting of L. The optimal L is achieved at 0.1, i.e., the popularity level is estimated with 10% of the overall cascades. The length of windows L is set to 0.1 for the following experiments on real-world datasets.

5.1.7 *Setting of number of clusters K*

In addition to the parameters in LSH, the CLSH method includes an additional parameter, namely the number of clusters K. Thus, K is varied from 2 to 5 with results as shown in Figure 6. Empirically, the results indicate that clustering cascades into 3 to 4 groups are associated with higher accuracy. With respect to experiments on real-world datasets, K is set to 3.

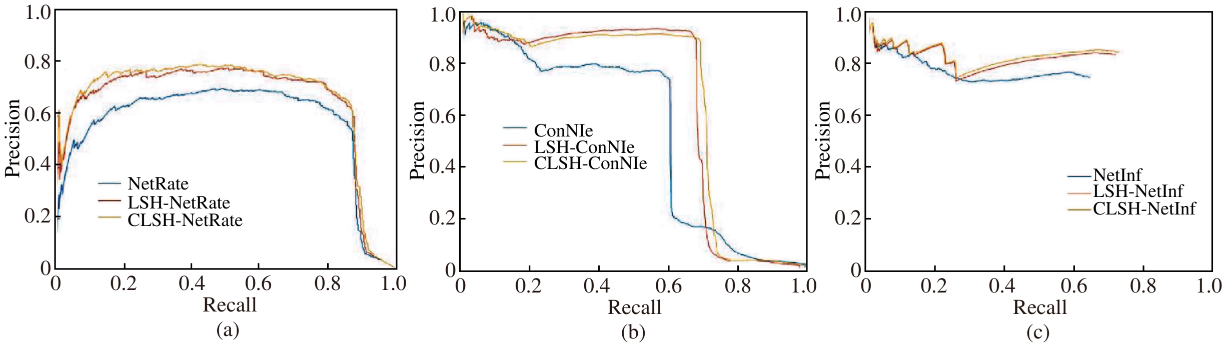


Figure 7 (Color online) Precision-recall curve with respect to MemeTracker dataset. (a) (C)LSH-NetRate; (b) (C)LSH-ConNIe; (c) (C)LSH-NetInf.

5.2 Experiments on real-world dataset

5.2.1 *Dataset*

US Patent dataset[1] by the National Bureau of Economic Research and MemeTracker dataset [24] are used to test the proposed (C)LSH method. The Sina microblog datasets are not used since the number of cascades in the dataset is too small. The Patent dataset consists of US patents in 25 years ranging from 1975 to 1999. In the dataset, each node represents an inventor. The top 400 inventors with most number of patents are selected in the experiment. The cascades are extracted as the diffusion tree of an innovative idea by following the citation relationship. The ground truth diffusion network is also constructed from the citation relationship between the inventors. An edge is added into the ground truth network if citation occurs once between two inventors. The MemeTracker dataset collects the on-line articles from Blogsphere and main-stream media sites. In the dataset, each node corresponds to a media sit, while the cascades correspond to the propagation of Memes. The top 500 sites with most number of articles are extracted in the experiments. The ground truth network is generated based on the hyperlink contained in the on-line articles by following the procedure detailed in a previous study [10].

With respect to US Patent datasets, the largest 850 cascades are used in the experiments while the largest 2000 cascades are used in the MemeTracker datasets.

5.2.2 *Algorithms*

The (C)LSH-NetRate, (C)LSH-ConNIe, and (C)LSH-NetInf as well as three original algorithms are used as baselines on the two datasets. With respect to all algorithms, it is assumed that the delay time follows the exponential distribution. With respect to (C)LSH-ConNIe and (C)LSH-NetInf, the parameter of exponential distribution is set to 0.5.

5.2.3 *Results*

Figure 7 plots the full precision-recall curve of by incorporating the proposed method to all the three baseline network inference algorithms on the MemeTracker dataset. The large margin between the proposed approach and the baselines clearly demonstrates the effectiveness of the proposed method. The CLSH-ConNIe achieves the best performance with 17.06% in F1 score and 21.68% in AUC improvement when compared to the best baseline method. The precision-recall PRcurve on Patent dataset is shown in Figure 8. The CLSH-NetInf performs the best with an even more significant improvement of 28.98% in F1 score and 62.27% in AUC.

Additionally, the performance of the proposed approaches is tested by varying the number of cascades in real-world datasets. The results are shown in Figures 9 and 10. The proposed method can infer the network structure even with inadequate cascades and results in a significant improvement to the baseline algorithms. The proposed (C)LSH achieves an improvement exceeding 10% with respect to the ConNIe

[1] http://www.nber.org/patents/.

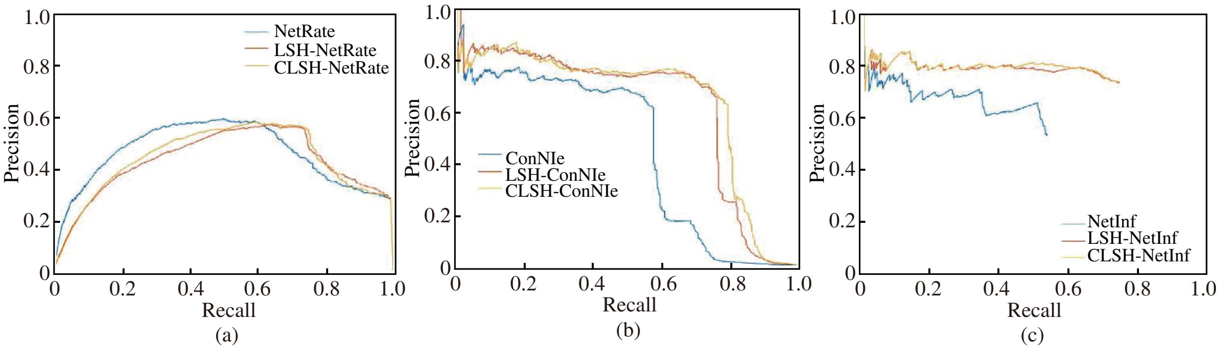


Figure 8 (Color online) Precision-recall curve with respect to Patent dataset. (a) (C)LSH-NetRate; (b) (C)LSH-ConNIe; (c) (C)LSH-NetInf.

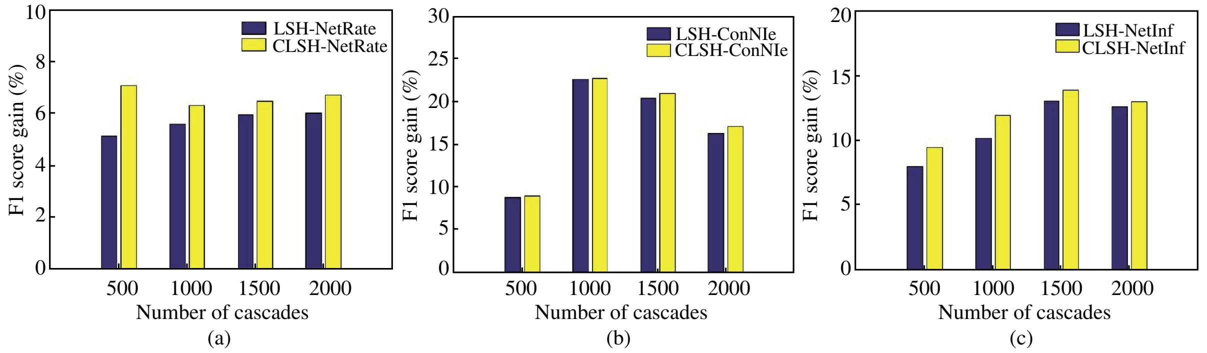


Figure 9 (Color online) Performance on MemeTracker dataset vs. the number of cascades. (a) (C)LSH-NetRate; (b) (C)LSH-ConNIe; (c) (C)LSH-NetInf.

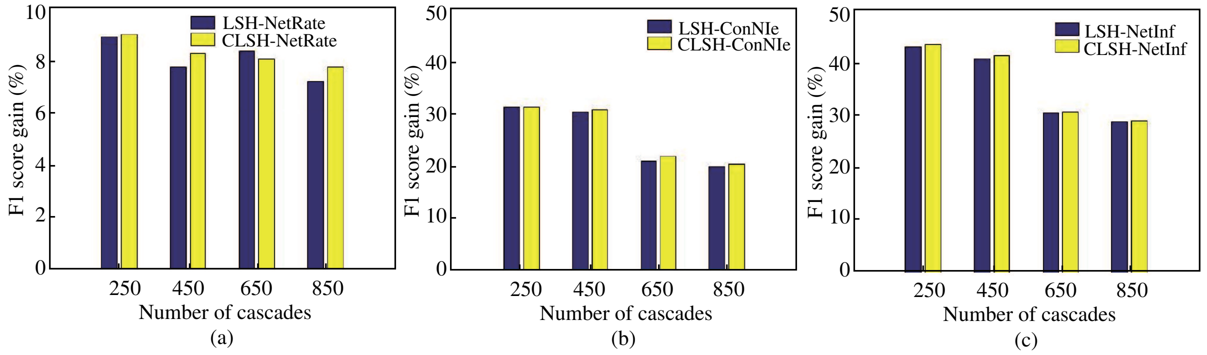


Figure 10 (Color online) Performance on Patent dataset vs. the number of cascades. (a) (C)LSH-NetRate; (b) (C)LSH-ConNIe; (c) (C)LSH-NetInf.

and NetInf on the MemeTracker datasets and an improvement exceeding 30% with respect to the Patent dataset in which the number of cascades is small. The extreme simplicity of applying the proposed method leads to significant improvements that are almost free. This suggests that it is a good idea to always try to implement the proposed to solve a network inference problem.

The popularity level heterogeneity inferred by the proposed CLSH algorithm is further visualized in Figure 11 that plots the popularity level of two clusters in MemeTracker and Patent datasets. The popularity level in MemeTracker is considerably similar to the pattern shown in Figure 1 of Sina microblog cascades where a Meme begins as popular initially and becomes stale as the propagation proceeds. Conversely, the new patent gradually gains popularity or remains unrecognized for a long time and becomes extremely popular afterwards. The estimation matches the empirical observation in Figure 1 of Patent cascades, and this validates the proposed method for the popularity level.

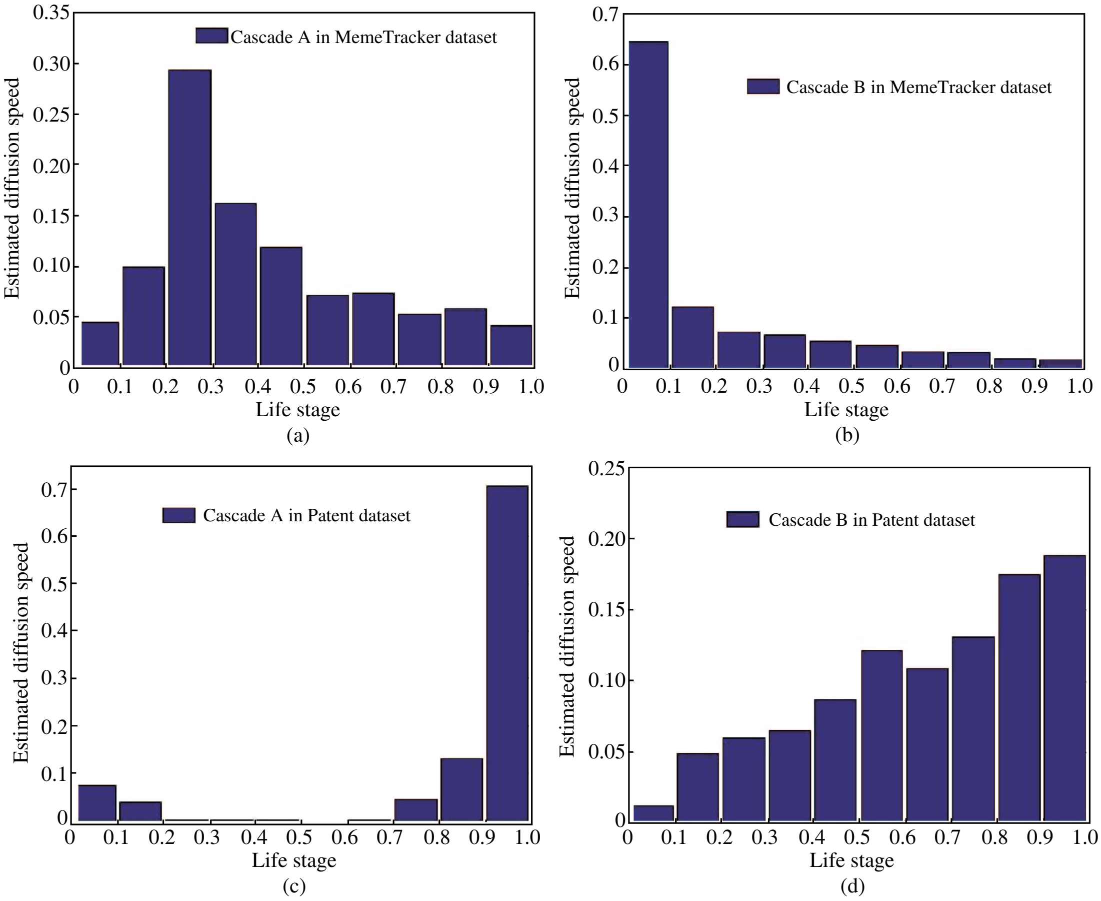


Figure 11 (Color online) The popularity level estimated by CLSH in MemeTracker and Patent datasets. (a) Cluster A in MemeTracker; (b) cluster B in MemeTracker; (c) cluster A in Patent dataset; (d) cluster B in Patent dataset.

6 Conclusion

In this study, the Life Stage Heuristics (LSH) and Clustered Life Stage Heuristic (CLSH) methods were proposed to improve the accuracy of network inference by incorporating the heterogeneity of influence strength in different stages of the diffusion processes. The proposed methods can be applied to almost all existing network inference algorithms to improve the accuracy of network inference without major changes in implementation and increases in running time. Specifically, NetInf, NetRate, and ConNIe algorithms are used as examples to demonstrate the power of the proposed approaches. Experiments on both synthetic and real-world datasets demonstrate the superiority of the proposed approaches.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61572041), Beijing Natural Science Foundation (Grant No. 4152023), and National High Technology Research and Development Program of China (863 Program) (Grant No. 2014AA015103).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Domingos P, Richardson M. Mining the network value of customers. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 2001. 57–66

2 Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, 2003. 137–146

3 Leskovec J, Adamic L A, Huberman B A. The dynamics of viral marketing. In: Proceedings of ACM Transactions on the Web, New York, 2007. 1–5

4 Adar E, Zhang L, Adamic L, et al. Implicit structure and the dynamics of blogspace. In: Proceedings of Workshop on the Weblogging Ecosystem, New York, 2004. 13: 16989–16995

5 Gruhl D, Guha R, Liben-Nowell D, et al. Information diffusion through blogspace. In: Proceedings of the International Conference on World Wide Web, New York, 2004. 491–501

6 Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, 2005. 177–187

7 Dong X L, Berti-Equille L, Srivastava D. Truth discovery and copying detection in a dynamic world. Proc VLDB Endow, 2009, 2: 562–573

8 Daneshmand H, Rodriguez M G, Song L, et al. Estimating diffusion network structures: Recovery conditions, sample complexity and soft-thresholding algorithm. In: Proceedings of International Conference on Machine Learning, Beijing, 2014. 793–801

9 Du N, Song L, Woo H, et al. Uncover topic-sensitive information diffusion networks. In: Proceedings of International Conference on Artificial Intelligence and Statistics, Scottsdale, 2013. 229–237

10 Du N, Song L, Yuan S, et al. Learning networks of heterogeneous influence. In: Proceedings of Neural Information Processing Systems Conference, Lake Tahoe, 2012. 2780–2788

11 Rodriguez M G, Balduzzi D, Schölkopf B. Uncovering the temporal dynamics of diffusion networks. In: Proceedings of International Conference on Machine Learning, Bellevue, 2011. 561–568

12 Rodriguez M G, Leskovec J, Krause A. Inferring networks of diffusion and influence. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012. 5: 21

13 Wang S, Hu X, Yu P S, et al. Mmrate: inferring multi-aspect diffusion networks with multi-pattern cascades. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2014. 1246–1255

14 Yang S H, Zha H. Mixture of mutually exciting processes for viral diffusion. In: Proceedings of International Conference on Machine Learning, Atlanta, 2013. 28: 1–9

15 Zhou K, Zha H, Song L. Learning social infectivity in sparse low-rank network using multi-dimensional hawkes processes. In: Proceedings of International Conference on Machine Learning, Atlanta, 2013. 31: 641–649

16 Dou P, Du S Z, Song G J. Inferring diffusion network on incomplete cascade data. In: Proceedings of International Conference on Web-Age Information Management, Nanchang, 2016. 325–337

17 Pasumarthi R K, Karthik S, Choure A, et al. Online network inference under dynamic cascade updates: A node-centric approach. In: Proceedings of International Conference on Communication Systems and Networks, Bangalore, 2014. 1–6

18 Rodriguez M G, Scholkopf B. Submodular inference of diffusion networks from multiple trees. In: Proceedings of International Conference on Machine Learning, Edinburgh, 2012. 1019–1028

19 Myers S A, Leskovec J. On the convexity of latent social network inference. In: Proceedings of Neural Information Processing Systems Conference, Whistler, 2010. 1741–1749

20 Yang J ,Leskovec J. Patterns of temporal variation in online media. In: Proceedings of International Conference on Web Search and Data Mining, New York, 2011. 177–186

21 Matsubara Y, Sakurai Y, Prakash B A, et al. Rise and fall patterns of information diffusion: Model and implications. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012. 6–14

22 Evans J D. Straightforward Statistics for the Behavioral Sciences. California: Brooks/Cole Publishing, 1996

23 Leskovec J, Chakrabarti D, Kleinberg J, et al. Kronecker graphs: an approach to modeling networks. J Mach Learn Res, 2010, 11: 985–1042

24 Leskovec J, Backstrom L, Kleinberg J. Meme-tracking and the dynamics of the news cycle. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, 2009. 497–506