CT LIS: Learning Influences and Susceptibilities through Temporal Behaviors

SHENGHUA LIU and HUAWEI SHEN, Institute of Computing Technology, Chinese Academy of Sciences, CAS Key Laboratory of Network Data Science & Technology, University of Chinese Academy of Sciences
HOUDONG ZHENG, Fuzhou University
XUEQI CHENG, Institute of Computing Technology, Chinese Academy of Sciences, CAS Key Laboratory of Network Data Science & Technology, University of Chinese Academy of Sciences
XIANGWEN LIAO, Fuzhou University

57

How to quantify influences between users, seeing that social network users influence each other in their temporal behaviors? Previous work has directly defined an independent model parameter to capture the interpersonal influence between each pair of users. To do so, these models need a parameter for each pair of users, which results in high-dimensional models becoming easily trapped into the overfitting problem. However, such models do not consider how influences depend on each other if influences are sent from the same user or if influences are received by the same user. Therefore, we propose a model that defines parameters for every user with a latent influence vector and a susceptibility vector, opposite to define influences on user pairs. Such low-dimensional representations naturally cause the interpersonal influences involving the same user to be coupled with each other, thus reducing the model's complexity. Additionally, the model can easily consider the temporal information and sentimental polarities of users' messages. Finally, we conduct extensive experiments on two real-world Microblog datasets, showing that our model with such representations achieves best performance on three prediction tasks, compared to the state-of-the-art and pair-wise baselines.

CCS Concepts: • **Information systems** → **Social networks**; **Web log analysis**; *Data mining*; • **Computing methodologies** → *Machine learning*;

Additional Key Words and Phrases: User behaviors, influence, susceptibility, time series

ACM Reference format:
Shenghua Liu, Huawei Shen, Houdong Zheng, Xueqi Cheng, and Xiangwen Liao. 2019. CT LIS: Learning Influences and Susceptibilities through Temporal Behaviors. *ACM Trans. Knowl. Discov. Data* 13, 6, Article 57 (October 2019), 21 pages.
https://doi.org/10.1145/3363570

This material is based upon work supported by the Strategic Priority Research Program of CAS under Grant XDA19020400, the Beijing National Science Foundation under Grant 4172059, and the National Science Foundation of China under Grants 61772498, 61872206, U1605251. This work is also supported by the Open Project of Key Laboratory of Network Data Science & Technology of CAS under Grants CASNDST201708 and CASNDST201606, and the Open Project of National Laboratory of Pattern Recognition at the Institute of Automation of CAS (201900041).
Authors' addresses: S. Liu (corresponding author), H. Shen (corresponding author), and X. Cheng, No. 6 Kexueyuan Road, Zhongguancun Haidian District, Beijing, China; emails: {liushenghua, shenhuawei, cxq}@ict.ac.cn; H. Zheng and X. Liao (corresponding author), College of Mathematics and Computer Science, Fuzhou University, China; email: houdongzheng@foxmail.com, liaoxw@fzu.edu.cn.

1556-4681/2019/10-ART57 $15.00
https://doi.org/10.1145/3363570

1 INTRODUCTION

Social network services generally allow users to post, forward, share, or "like" a piece of information; to comment on a product or service; or to "check in" at a place of interest (POI). All the above behaviors are temporal and can be grouped by those objects, such as pieces of information, products, and places. The group associated with an object is known as a temporal cascade, which is a sequence of users' behaviors. Thus, a temporal cascade contains users' actions with regard to a specific object. Since such actions are publicly visible, and since the system purposefully feeds the information to related users or communities, then users can influence each other [26, 38], forming cascades, which is similar to a contagion of behaviors [46].

Scholars have applied the concept of contagion to study viral marketing [34, 42], influence maximization [8, 14, 19, 39], and how opinion forms [6, 56, 58]. These studies and applications need a way to know the causality of who influences whom, as well as accurate values of influences. In addition, the influences are usually different, depending on who the actors and recipients of the action are [1]. Therefore, accurately quantifying the interpersonal influences involved is fundamental to a proper understanding of viral marketing, influence maximization, and understanding opinion formation.

In recent works, interpersonal influences are estimated by machine learning or statistical techniques from the observed data. However, the modeling of users' influences should address the following problems.

Overfitting. The model should consider the sparse of user behaviors in cascades, avoiding free parameters for seldom observed behaviors [55]. Saito et al. [44] defined a success probability that a user would cause another user to act. The influences were learned as probabilities by maximizing likelihood. After that, the influences were described with transmission rates between user pairs [21, 25]. The rates indicated the speed at which the information is passed on, and infection times were also considered in the model. However, those models learned interpersonal influences by directly defining a free parameter on each user pair—either propagation probability or transmission rate. Such pair-wise parameters do not consider dependency, especially when those influences originate from or act on the same user. As shown in Figure 1, we can observe information passed through cascades (a, c, e, f) and (a, b, e, d, f). The interpersonal influence between c and d cannot be learned, since they do not appear in the same cascade with information passed between them. In such an extreme case, c, d, and e can form a triangle relationship, which is usually observed in social network. Nevertheless, previous works assigned zero or some empirically small constant as interpersonal influence, implying that information would never or seldom be passed between them. Therefore, those models suffer from overfitting problem in sparse data, because of how they define their model parameters, resulting in inaccurate estimation.

Unavailable causality relationship. Goyal et al. [26] estimated the influences as propagation probabilities, or the likelihood that the information would be passed on. These probabilities were learned by means of a Bernoulli or Jaccard Index model, relying on the causality data of who influences whom. However, whereas data on the times when users are influenced or infected are usually available, data on the causality are often unavailable.

Temporal information. Temporal information of when users act is useful to measure the influence [38]. A quickly infected user may be easily susceptible, reflecting the influences between the relevant users. NetInf [22, 23] considered the temporal effect between the infection of two users with an exponential model and a power-law model. Nevertheless, those works used many free parameters, suffering overfitting.

Therefore, we propose a concise model by defining model parameters for individual users as opposed to user pairs, specifically users' *influence vectors* and *susceptibility vectors*. For clarity, we

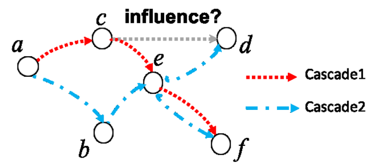


Fig. 1. Previous pair-wise work gave an overfitting influence to (c, d).

use *interpersonal influence* to indicate the likelihood that information will be passed between two people, and *influence vector* to indicate a unilateral influence from a user and *susceptibility vector* is a unilateral susceptibility of a user. The interpersonal influences are modeled by the product of the influence vector from one user and the susceptibility vector from another. In addition, we use *hazard function* in survival theory to model the dynamics of interpersonal influences by the elapsed time from originating time.

The experiments on synthetic data validate the effectiveness of our learning algorithms. And on two-real world datasets, ours achieves the best performance in three prediction tasks.

In summary, our "CT LIS" has the following advantages:

— *Dependency awareness:* CT LIS models the interpersonal influences by the product of the influence vector from one user and the susceptibility vector from another. Thus, when two interpersonal influences are sending from the same user, the influences can be coupled together by the shared influence vector. Similarly, when two interpersonal influences are received by the same user, the influences can be coupled together by the shared susceptibility vector.
— *Conciseness:* our model uses fewer parameters $O(n \cdot D)$, instead of $O(n^2)$ parameters for user pairs to represent users' influence, where n is the number of users, and D is the dimension of influence and susceptibility vectors. Using a less complex model offers an advantage in modeling sparse data.
— *Effectiveness:* our model achieved the best performance on two real-world Microblog datasets for three prediction tasks on: next infected user, cascade size, and "who will be retweeted" (see Figure 2).
— *Unification of temporal and categorical information:* our model proposes a unified framework based on temporal model, to make use of infection time, and categorical attributes, like sentiments and topics of review text.

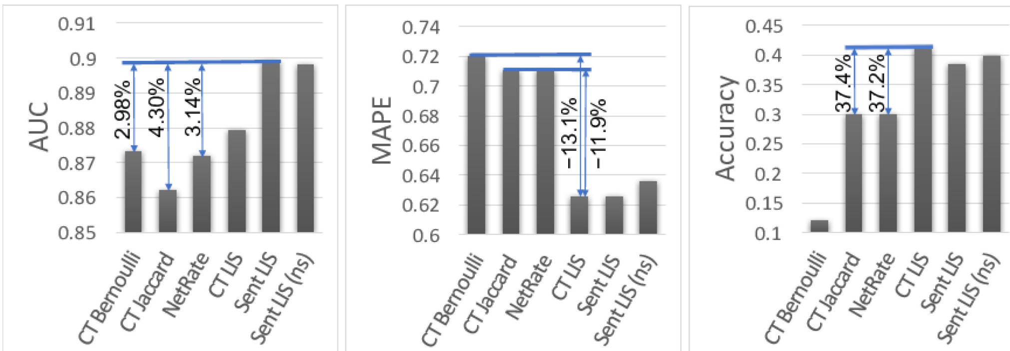
At last, with the learned representations of users, two groups of users, the "primary influential" and the "secondary influential," are located, and their interesting behavior patterns are analyzed.

The rest of the article is organized as follows. Section 2 surveys the existing and related work, and the motivation and model are described in Section 3, including the parameter learning algorithm. The experiments and analysis are reported in Section 4, and Section 5 provides conclusions.

2 BACKGROUNDS

2.1 Related Works

The previous work on cascade dynamics was studied based point-process or influence models. The authors in [18, 45, 63] used the variants of Poisson process to predict cascade size. Hawkes process-based work [4, 40, 57] was also studied to model the dynamics. [52, 54] proposed to use Recurrent Neural Network to predict the next point and time. Moreover, a recent work SMF [29] proposed to predict events (e.g., retweet) between users and objects by formalizing a dynamic graph and factorizing with seasonality. Those works do not explicitly learn users' influences.



(a) Our models outperform competitors on prediction of next infected user.

(b) Our results achieve the lowest error on cascade size prediction.

(c) Our models outperform competitors on prediction of "who will be retweeted".

Fig. 2. Our models are CT LIS, and its variants: Sent LIS and Sent LIS(ns). The area under ROC curve (AUC) is used as a measure for next infected user prediction. Mean absolute percentage error (MAPE) is used as a measure for cascade size prediction, and a good model will have a small MAPE. Top-one accuracy is used to measure the prediction accuracy of "who will be retweeted."

2.1.1 Learning Influences without Temporal Information. In terms of influence models, scholars estimated interpersonal influences and their relationship to information propagation. Some of them made efforts to extract features that are related to propagation probability and learned from the observed information cascades [1, 2, 10]. Tang et al. [49] proposed a topic factor graph (TFG) to model the generative process of the topic-level social influence on large networks. [36] proposed a probabilistic factor graph to model the direct and indirect influences between adjacent and non-adjacent users of heterogeneous networks. Zhang et al. [62] learned influence from partial cascades. Saito et al. [44] learned the propagation probability between neighbors of a directed network, using the orders of users become infected. Goyal et al. [26] proposed to estimate the interpersonal influences, using assumptions from the Bernoulli model and Jaccard Index separately. NetInf [22] modeled propagation probability, using exponential and power-law incubation time separately to infer the underlying network. Most of the previous works directly model interpersonal influence as separate parameters, making model complex. LIS [53] proposed to use influence and susceptibility to model each user, which reduced the model parameters.

2.1.2 Learning Influences with Temporal Information. Moreover, a series of work learned interpersonal influence with survival model to consider temporal information, as well as inferring underlying networks. NetRate [21] used survival theory to model the transmission rate between every user pair, which was viewed as an edge weight of the influence network. [24] then modeled the hazard rate in a survival model with additive and multiplicative risks separately to improve the performance of cascade size prediction (CSP). Afterwards, InfoPath [25] was proposed to learn time-varying transmission rates for user pairs as the edge weights of the hidden dynamic network. The distribution of content topics has also been considered [15]. Taken together, these methods work in a pair-wise manner, i.e., they learned the propagation probability between pairs of users. This approach is fundamentally different from the proposed method proposed in this article, which focuses on inferring user-specific influence and susceptibility from historical cascades. Also influence representations on sentimental polarities can be learned in our model.

2.1.3 Modeling Sentiments in Information Propagation. In a field of information propagation, sentimental influence and opinion formation were studied by many researchers. [32, 59, 60] experimentally showed that users' sentiments were influenced by that of others surrounding them on LiveJournal dataset and Facebook dataset separately. Bae and Lee [3] used Granger causality analysis to show that sentiment change of audiences were related to the landscape of popular users in Twitter. TASC [37] showed that how the dynamic influence of opinions helped to improve sentiment classification by semi-supervision.

Many models were proposed to model opinion dynamics, including Sznajd model [48], Deffuant model [12], and Hegselmann and Krause model [27], which produced agreeing results. Moreover, Rodrigues and Costa [43] extended Sznajd model to complex networks. And Tu et al. [50] found that a large clustering coefficient helped the opinion dynamics to reach consensus. Deffuant et al. [12] modeled the process of opinion dynamics that randomly select two users, and change their opinions to reduce the difference. Hestenes [28] modeled to change users' opinions according to the arithmetic average of that of their neighbors, and Fortunato et al. [17] extended the model with multi-dimensional opinion vector, instead of a scalar opinion value. Besides, Suchecki et al. [47] studied Voter model in scale-free network, small-world network, and random network.

The studies in opinion dynamics then focused more on the convergence. A recent work [6] by Bindel et al. discovered that traditional models including DeGroot model [13] finally converged to a state of consensus under a set of general conditions, while it is rare in real opinion dynamics. Hence, it proposed to model with users' intrinsic beliefs in a game theory, which counterbalanced the opinions at Nash equilibrium. In addition, Gionis et al. [19] studied the overall positive opinion maximization problem, adopting the game model [6] of opinion dynamics. De at al. [11] modeled that a user's opinion was generated from her latent opinion distribution, based on self-excited Hawkes process influenced by her neighbors.

The above works are mostly on opinion dynamics and maximization, assuming the sentimental influences between connected users were equal. It does not confirm to our observations in real life, in which a minority of influential users infect an exceptional number of their peers [30], and a mass of users are easily influenced. As far as we know, an effective method for learning sentimental influences is still needed.

2.2 Survival Analysis Model

We introduce the prior knowledge of the Survival Analysis Model [24, 33]. Define the infection time T of a user acting on a target as a continuous random variable, with $T \in [0, \infty)$. Let $f(t)$ and $F(t)$ denote the probability density function (PDF) and the cumulative density function (CDF), respectively. The probability $Pr(T \leq t) = F(t)$. So the probability of a user not taking the action until time t is defined by the survivor function:

$$S(t) = Pr(T \geq t) = 1 - F(t) = \int_t^{\infty} f(x)dx.$$

A hazard function $h(t)$ is defined as the instantaneous infection rate in time interval $[t, t + \varepsilon)$, where ε is an infinitesimal elapsed time, given that a user survives until time t:

$$h(t) = \lim_{\varepsilon \to 0} \frac{Pr(t \leq T < t + \varepsilon | T \geq t)}{\varepsilon} = \frac{f(t)}{S(t)}.$$

Noticing that $f(t) = -S'(t)$ and $S(0) = 1$, the survivor function can be expressed as

$$\ln S(x) = -\int_0^t h(x)dx. \tag{1}$$

The survival analysis model was originally used for modeling the life time of an object [33], and we will show later that it helps us to consider temporal information in a cascade.

3 LEARNING USERS' INFLUENCES

A temporal cascade C_i for actions on a target i is defined as a list tuples:

$$C_i = \{(v_1, t_1), (v_2, t_2), \ldots, (v_N, t_N)|t_1 \leq t_2 \leq \cdots \leq t_N\},$$

where v_i is the user who takes the action at time t_i, and N is the total number of acting users, i.e., the cascade size. To simplify the description of our model, we do not specify the identity i of a target or message, until we establish the overall objective function at the end. In applications, C can be a sequence of users retweeting a message in Twitter; or a sequence of users clicking "like" to a webpage or a photo in Facebook. Thus, we generally can observe a large number of cascades for different objects in each application.

The network of causality of who influences whom is hidden and not available in most cases. As a substitute for a causality network, a social network was used sometimes, which is still arguable in academia. Moreover, social networks in blogs, Yelp, YouTube, and shopping are not available either.

To make our model more general, we assume that any pair of users can have an interpersonal influence, or probability of passing information between them, as most existing works do [21]. In such a case, a very small or zero value of influence can capture the underlying disconnections of the user network, and vice versa. Note that a social network is also easily applied in our model by setting influences between disconnected edges to zero, instead of a parametric function.

Therefore, our problem can be informally described as follows for simplicity, while the formal definition is introduced later in Section 3.1:

Informal Problem 1 (learning users' influences). *Given a set of cascades $\mathbb{C} = \{C_1, C_2, \ldots\}$.*

- *Learn the influences between users in form of model parameters.*
- *to optimize the likelihood of observing such cascades under our model.*

3.1 Our Proposed Model

We generally consider a user v_i passes a message at a time under the influences, as the life time that v_i is "alive" until v_i becomes infected and passes the message in survival analysis (see Section 2.2). The hazard $h(t)$ defines the "speed" that v_i is "dying," which can be related to the influences continuously giving from surrounding infected users, and their infection time, as we assume that a user contributes to influences once the user gets infected.

To define hazard $h(t)$, our model represents a user v_i with two non-negative D-dimensional vectors I_i and S_i, which stand for influence representation and susceptibility representation, respectively. D is a tuning parameter, which can be interpreted as how many hidden channels/dimensions that a user can influence another. For example, if $D = 4$, $I_u = [1, 1, 0, 0]$ and $S_u = [1, 1, 0, 0.5]$ for a user u. Then user u has influence of level 1 in the first two channels (say spots and musics), and no influence in the last two (say politics and technologies).

To consider sentiments, we use a representation vector for each sentiment class, and stack them into a matrix, i.e., matrix $\mathbf{I}_i$ for user's influence and matrix $\mathbf{S}_i$ for user's susceptibility. Each row of a matrix is the representation vector on a sentimental class. Thus, $\mathbf{I}_i$ and $\mathbf{S}_i$ are $K \times D$ dimensional matrices, where K is the number of sentiment classes.

Let one-of-K vector o indicate sentiment class, where only one element in vector o can be 1, and the others are zero. Note that if $K = 1$ and $o = 1$, the model degenerates back to vector representations without sentimental information. We then introduce our model in a general form with sentiment o in the following sections.

First, we consider a case of an infected user v_j influencing user v_j. For a cascade with sentiment class o, the transmission rate function $\phi(\cdot)$ from users v_j to v_i, is defined by equation:

$$\phi(\mathbf{I}_j, \mathbf{S}_i, o) = 1 - \exp\Big\{ - o^T \mathbf{I}_j \mathbf{S}_i^T o \Big\}, \tag{2}$$

where matrix $\mathbf{I}_j$ and matrix $\mathbf{S}_i$ are parameters that capture the influence of user v_j and the susceptibility of user v_i, respectively. The transmission rate function (2) indicates the likelihood of successful passing of information between them. We use an exponential function to scale the transmission rate between 0 and 1 for regularization.

Example 1 (Different Transmission Rates). We will show an example of how influence and susceptibility vectors decide different transmission rates, i.e., interpersonal influences. Let $D = 4$, $K = 1$ and $o = 1$. Then, $I_u = [1, 1, 0, 0]$ for user u. $S_a = [1, 1, 0, 0]$ for user a and $S_b = [0, 0, 1, 1]$ for user b. So the transmission rate $\phi(I_u, S_a, o)$=$1 - e^{-I_u S_a^T} = 1 - e^{-2}$ shows more interpersonal influence between u and a, than that between u and b, i.e., $\phi(I_u, S_b, o) = 0$. Influential user may only have influences in several channels as well as susceptibilities, which agrees with the observed in real world. Therefore, only if two users match in each channel of influence vector and susceptibility vector, they can have strong interpersonal influence; otherwise, their influence is weak. That is how a user's actions can have more influence on one user, while less influence on another.

To simplify, let $\mathcal{H}_{ji}$ denote the set of parameters $\{\mathbf{I}_j, \mathbf{S}_i, o\}$. With a transmission rate $\phi(\mathcal{H}_{ji})$, we define the hazard function of the Survival Analysis Model for user v_i at time t, under the influence from v_j, as follows:

$$h(t|t_j; \phi(\mathcal{H}_{ji})) = \phi(\mathbf{I}_j, \mathbf{S}_i, o) \frac{1}{t - t_j + 1}, \tag{3}$$

where $t - t_j + 1$ depicts the hazard function monotonously decaying with the time elapsed from t_j. Adding 1 avoids an unbounded hazard rate due to a zero or infinitesimal value of $t - t_j$. Since Equation (3) holds only when $t \geq t_j$, we define the hazard rate $h(t|t_j; \phi(\mathcal{H}_{ji})) = 0$, when $t < t_j$, namely, user v_j has not been infected at time t. Moreover, as mentioned earlier, we can consider social network as another constraint by defining hazard function $h(t|t_j; \phi(\mathcal{H}_{ji})) = 0$, if user v_i and user v_j are not connected.

Given the survivor function (1) in the Survival Analysis Model, the survivor function $S(t|t_j; \phi(\mathcal{H}_{ji}))$ that user v_i survives longer than t satisfies

$$\begin{aligned} \ln S(t|t_j; \phi(\mathcal{H}_{ji})) &= -\int_0^t h(x|t_j; \phi(\mathcal{H}_{ji})) dx \\ &= \phi(\mathbf{I}_j, \mathbf{S}_i, o) \cdot \ln(t - t_j + 1). \end{aligned} \tag{4}$$

Finally, given that influential user v_j takes action or becomes infected at time t_j, the PDF of user v_i happening (acting on the target) at time t is

$$f(t|t_j; \phi(\mathcal{H}_{ji})) = h(t|t_j; \phi(\mathcal{H}_{ji})) S(t|t_j; \phi(\mathcal{H}_{ji})).$$

Since every previously infected users can output a probability to infect user v_i, we combine them together by assuming that a user is only infected by one of the previously infected users [24], the likelihood of user $v_i, i > 1$ being infected at time t_i in a cascade is

$$\begin{aligned} f(t_i|\boldsymbol{t}; \phi(\boldsymbol{\mathcal{H}})) &= \sum_{j:t_j<t_i} f(t_i|t_j; \phi(\mathcal{H}_{ji})) \prod_{k\neq j, t_k<t_i} S(t_i|t_k; \phi(\mathcal{H}_{ki})) \\ &= \sum_{j:t_j<t_i} h(t_i|t_j; \phi(\mathcal{H}_{ji}) \cdot \prod_{k:t_k<t_i} S(t_i|t_k; \phi(\mathcal{H}_{ki})). \end{aligned} \tag{5}$$

Thus, given that user v_1 takes the first action at time t_1, the joint likelihood of observing the whole cascade is

$$f(\boldsymbol{t} \setminus t_1|t_1;\phi(\boldsymbol{\mathcal{H}})) = \prod_{i>1} \sum_{j:t_j<t_i} h(t_i|t_j;\phi(\mathcal{H}_{ji})) \cdot \prod_{k:t_k<t_i} S(t_i|t_k;\phi(\mathcal{H}_{ki})).$$

To consider the negative cases, we define a time window of our observation for cascade C. The end of the time window is t_E, and $t_E > t_N$. The users who do not act until t_E are survivors under the influence of infected users. Thus, the probability of a negative case for the survival of user v_l is

$$S(t_E|\boldsymbol{t};\phi(\boldsymbol{\mathcal{H}})) = \prod_{i:t_i\le t_N} S(t_E|t_i;\phi(\mathcal{H}_{il})).$$

Considering the negative cases, the log-likelihood of a cascade is

$$\ln \mathcal{L}(\mathbf{I},\mathbf{S};o) = \sum_{i>1} \ln\left(\sum_{j:t_j<t_i} \phi(\mathbf{I}_j,\mathbf{S}_i,o)\frac{1}{t_i-t_j+1}\right) - \sum_{i>1}\sum_{k:t_k<t_i} \phi(\mathbf{I}_k,\mathbf{S}_i,o)\cdot \ln(t_i-t_k+1) - \sum^{L} \mathbb{E}_{v_l\sim P(u)}\left[\sum_{j=1}^{N} \phi(\mathbf{I}_j,\mathbf{S}_l,o)\cdot \ln(t_E-t_j+1)\right].$$

It is observed that a message is usually passed by a few users. Compared to the overall population in a social network, most of users are survived. Consider that negative pairs are formed by any infected user and any survivor, i.e., the total size of negative cases is the number of infected users multiply the number of survivors. Hence, we can have very unbalanced negative and positive cases. The imbalance may mislead the optimization directions, resulting in loss of attention on positive cases. Besides, a large number of negative cases slow down optimization algorithm, limiting the scalability. Thus, negative sampling strategy [41] is necessarily applied.

The frequency of a user infected in all cascades indicates how easily the user could become infected again. Observing a frequently infected user who survives in a cascade can provide more information to the likelihood. So, sampling survivors of a cascade from the distribution of their infected frequencies is a better choice: sample L survival users according to distribution $P(u) \propto R_u^{3/4}$, where R_u is the total infection frequency of user u. Negative pairs are then formed by each of infected users in a cascade and each sampled survivor. Note that sampling of negative cases is repeated in every optimization iteration, so the samples can honor the expectation of infection frequency.

Finally, the formal optimization problem of learning users' influence representations and susceptibility representations is

Problem 1 (learning users' influences and susceptibilities).

$$\min_{\mathbf{I},\mathbf{S}} -\sum_{C} \ln \mathcal{L}^c(\mathbf{I},\mathbf{S};\boldsymbol{o}^c) \tag{6a}$$

$$s.t.\ \boldsymbol{I}_{ki} \ge \mathbf{0}, \boldsymbol{S}_{ki} \ge \mathbf{0}, \forall k,i, \tag{6b}$$

where superscript c is used to identify the values or functions that are related to cascade C.

The log-likelihood of all the observed cascades is summarized in the above objective function. Since transmission rate is function of product of $\mathbf{I}_j$ and $\mathbf{S}_i$ in Equation (2), we constraint elements in matrices to be non-negative to avoid another equivalent solution of $-\mathbf{I}_j$ and $-\mathbf{S}_i$. Note that we do not use negative value in influence matrix to indicate susceptibility, since we model each user with both influence and susceptibility matrices. Besides, for general purposes, different sentiment classes in our model, use different sets of vectors, which are stacked as matrices. So we cannot use positive and negative values in $\mathbf{I}_j$ and $\mathbf{S}_i$ to represent different sentiments either. Nevertheless, non-negative values in influence and susceptibility matrices can give the interpretability to our model, namely, the extent of how a user is influential and susceptible to others.

3.2 Optimization

As for learning the model parameters, we maximize the likelihood of recognizing the successful passing of information as a positive case, and an unsuccessful or unobserved passing as a negative case. Note that the number of negative cases is far more than that of positive ones. So considering all the negative cases prevents our model from both being applied to a very large data and balancing the positive and negative cases. Thus, in each iteration, we use a negative sampling strategy to approximate the expectation of the negative case distribution. The details are provided as follows.

The gradients of the transmission rate function on matrix $\mathbf{I}_v$ and matrix $\mathbf{S}_u$ are

$$\frac{\partial \phi(\mathbf{I}_v, \mathbf{S}_u, o)}{\partial \mathbf{I}_v} = (1 - \phi(\mathbf{I}_v, \mathbf{S}_u, o))oo^T S_u,$$
$$\frac{\partial \phi(\mathbf{I}_v, \mathbf{S}_u, o)}{\partial \mathbf{S}_u} = (1 - \phi(\mathbf{I}_v, \mathbf{S}_u, o))oo^T I_v.$$

The gradients are $K \times D$ matrices. Only the kth row in each matrix has a non-zero gradient, when a cascade belongs to the kth sentiment class, i.e., $o_k = 1$.

As the negative cases for a cascade are repeatedly sampled in every iteration, we define $[\mathbb{V}_s^c]_\tau$ as the set of negative users in the τth iteration of the algorithm for cascade C:

$$[\mathbb{V}_s^c]_\tau = \{v_l \sim P(u)\}_L,$$

where L is the set size.

The influence matrix $\mathbf{I}_v$ of user v only appears in the likelihood of a cascade c, if v becomes infected, i.e., $t_1^c \leq t_v^c \leq t_N^c$ in cascade c. The gradient of the log-likelihood (6a) on matrix $\mathbf{I}_v$ is

$$g_{I_v} = -\sum_c \mathbf{1}\left(t_v^c \leq t_N^c\right) \frac{\partial \mathcal{L}^c(\mathbf{I}, \mathbf{S}; \boldsymbol{o}^c)}{\partial \mathbf{I}_v}.$$

In terms of susceptibility matrix $\mathbf{S}_v$, if user v is also infected in a cascade c, i.e., $t_1^c < t_v^c \leq t_N^c$, or user v is in a sampled negative pair, the gradient of the likelihood of a cascade c is non-zero; otherwise, the gradient is always zero. Therefore, the gradient of the summation of all cascades in objective (6a) on matrix $\mathbf{S}_v$ is

$$g_{S_v} = -\sum_c \mathbf{1}\left(t_1^c < t_v^c \leq t_N^c\right) \frac{\partial \mathcal{L}^c(\mathbf{I}, \mathbf{S}; \boldsymbol{o}^c)}{\partial \mathbf{S}_v}$$
$$+ \sum_c \mathbf{1}\left(v \in \left[\mathbb{V}_s^c\right]_\tau\right) \cdot \sum_{j=1}^{N^c} (1 - \phi(\mathbf{I}_j, \mathbf{S}_v, o^c)) \cdot \ln\left(t_E^c - t_j^c + 1\right) o^c o^{cT} \mathbf{I}_j,$$

where $\mathbf{1}(\cdot)$ is an indicator function, with an output of 1 if the argument is true, and 0 otherwise. The gradients g_{I_v} and g_{S_v} are $K \times D$ matrices.

We use stochastic gradient descent (SGD) framework over shuffled mini-batches for optimization efficiency, setting 12 as the size of mini-batch. To consider non-negative constraint on parameters, projected gradient (PG) [35] is used to adjust the gradients.

Let the parameter updates be $\Delta\mathbf{I}_v$ and $\Delta\mathbf{S}_v$ for each user v. And matrices $\mathbf{I}, \mathbf{S} \in \mathbb{R}^{K\times(D\cdot M)}$ are separately the concatenations of $\mathbf{I}_v$ and $\mathbf{S}_v$ for all users v. M is the user count as defined previously. Thus, the updates will be reduced by a rate $0 < \beta < 1$, namely, $\beta\Delta\mathbf{I}_v$ and $\beta\Delta\mathbf{S}_v$, if the following condition does not hold:

$$\mathcal{O}([\mathbf{E}]_{\tau+1}) - \mathcal{O}([\mathbf{E}]_{\tau}) \leq \sigma \cdot Tr(\nabla\mathcal{O}([\mathbf{E}]_{\tau})^T([\mathbf{E}]_{\tau+1} - [\mathbf{E}]_{\tau})), \tag{7}$$

where $[\cdot]_\tau$ means the parameter in the τ-th iteration. With $\mathbf{E} = \{\mathbf{I}, \mathbf{S}\} \in \mathbb{R}^{K\times 2DM}$, $\mathcal{O}(\mathbf{E})$ is the simplified representation of objective function (6a). $Tr(\cdot)$ is the trace of a matrix, and σ is a constant between 0 and 1. We set $\sigma = 0.01$ and $\beta = 0.1$ as PG suggested.

Moreover, since deciding learning rate is not trivial, so we can choose one of well-known methods, Adadelta [61]. Let ρ be decay rate and ϵ be a small constant. The accumulate gradients in Adadelta are

$$\begin{aligned}
E[g_{I_v}^2]_\tau &= \rho E[g_{I_v}^2]_{\tau-1} + (1-\rho)[g_{I_v}^2]_\tau, \\
E[g_{S_v}^2]_\tau &= \rho E[g_{S_v}^2]_{\tau-1} + (1-\rho)[g_{S_v}^2]_\tau, \\
E[\Delta\mathbf{I}_v{}^2]_\tau &= \rho E[\Delta\mathbf{I}_v{}^2]_{\tau-1} + (1-\rho)[\Delta\mathbf{I}_v{}^2]_\tau, \\
E[\Delta\mathbf{S}_v{}^2]_\tau &= \rho E[\Delta\mathbf{S}_v{}^2]_{\tau-1} + (1-\rho)[\Delta\mathbf{S}_v{}^2]_\tau.
\end{aligned}$$

And with the definition of function $RMS[x]_\tau = \sqrt{E[x^2]_\tau + \epsilon}$, the update values are calculated as

$$\begin{aligned}
[\Delta\mathbf{I}_v]_\tau &= -\frac{RMS[\Delta\mathbf{I}_v]_{\tau-1}}{RMS[g_{I_v}]_\tau}[g_{I_v}]_\tau, \\
[\Delta\mathbf{S}_v]_\tau &= -\frac{RMS[\Delta\mathbf{S}_v]_{\tau-1}}{RMS[g_{S_v}]_\tau}[g_{S_v}]_\tau.
\end{aligned}$$

Therefore, the algorithm of learning users' sentimental influences is listed in Algorithm 1. Let the project function $\psi(x)$ be defined as projecting x into non-negative space, namely, $\psi(x) = 0$ if $x < 0$; otherwise $\psi(x) = x$. We use $\rho = 0.95$ and $\epsilon = 10^{-6}$ in the algorithm as Adadelta suggested.

Another recommended method, Adam [31], extends Adadelta with additional accumulated first-order moment of gradients, which might bring benefit to our optimization. We can easily apply Adam algorithm to the optimization by adding the extra smoothing gradients. However, the comparison between them is out of our contributions.

4 EXPERIMENTS

To evaluate our model, we used Microblog data, in which users' activity in passing messages is publicly available. The temporal cascades were extracted for all the messages in the data. We analyzed the learned influence representations and susceptibility representations. Finally, the article demonstrates the performance of our model with the well-known applications, in comparison to the state-of-the-art models.

4.1 Data

Synthetic data: To validate whether the proposed algorithm could obtain good estimation of latent influence and susceptibility vectors, we generate synthetic data for evaluation. First, we generate a synthetic diffusion networks using BA (Barabási–Albert) model [5]. We use 5-dimensional vectors for user's influence and susceptibility, and sample I_u and S_u from $f(x) = 1/2\sqrt{x}, x \sim U(0, 0.1)^5$, where $U(0, 0.1)$ is a uniform distribution between 0 and 0.1. We then generate cascades over the diffusion network according to our model. The first infected user is randomly selected for a cascade. In terms of times, we evenly set time ticks in a time unit starting from $t = 0$ for every cascade generation. And then we can simulate the processing according to our survival model, and flip a

ALGORITHM 1: Algorithm of learning users' sentimental influences.

Given $0 < \rho, \beta < 1$, constants σ and ϵ;
  initialized parameters $\mathbf{I}_v$ and $\mathbf{S}_v$ for each user v;
  Cascade set $\mathbb{C}$.

Iteration index $\tau := 0$;
$E[g^2_{I_v}]_0, E[g^2_{S_v}]_0, E[\Delta\mathbf{I}_v{}^2]_0, E[\Delta\mathbf{S}_v{}^2]_0 = 0$;
repeat
  Randomly shuffle $\mathbb{C}$;
  Split $\mathbb{C}$ into groups by mini-batch size;
  for each group **do**
    Compute gradients $[g_{I_v}]_\tau, [g_{S_v}]_\tau$;
    Accumulate gradients and updates:
      $E[g^2_{I_v}]_\tau, E[g^2_{S_v}]_\tau, E[\Delta\mathbf{I}_v{}^2]_\tau, E[\Delta\mathbf{S}_v{}^2]_\tau$;
    Parameter update values:
      $[\Delta\mathbf{I}_v]_\tau, [\Delta\mathbf{S}_v]_\tau$
    Update $[\mathbf{I}_v]_{\tau+1} = \psi([\mathbf{I}_v]_\tau + [\Delta\mathbf{I}_v]_\tau)$;
    Update $[\mathbf{S}_v]_{\tau+1} = \psi([\mathbf{S}_v]_\tau + [\Delta\mathbf{S}_v]_\tau)$;
    while not Condition (7) **do**
      decreasing update values:
        $[\Delta\mathbf{I}_v]_\tau = \beta[\Delta\mathbf{I}_v]_\tau, [\Delta\mathbf{S}_v]_\tau = \beta[\Delta\mathbf{S}_v]_\tau$
      Update $[\mathbf{I}_v]_{\tau+1} = \psi([\mathbf{I}_v]_\tau + [\Delta\mathbf{I}_v]_\tau)$;
      Update $[\mathbf{S}_v]_{\tau+1} = \psi([\mathbf{S}_v]_\tau + [\Delta\mathbf{S}_v]_\tau)$;
    end while
    $\tau := \tau + 1$
  end for
until parameters converged, or maximum epoch.

coin based on the likelihood (5) at every time tick. Finally 200 cascades are generated for 799 users in the diffusion network, and the total length of generated cascades are 15,343.

Since Microblog Sina Weibo,[1] the biggest Microblog website in China, is the best available data source for observing information diffusion. Thus, we sampled from Weibo database twice in 2013 and 2016 as two datasets for evaluations, named WB2013 and WB2016:

Microblog data WB2013: We initially sampled about 315.6 million records including posting, retweeting, and mentioning of messages between November 1, 2013 and February 28, 2014 from the timelines of 312,000 users from the Sina Weibo database. Since an emotion is a good way to determine the message's sentiment label [7, 20], we filtered the messages with frequently used emoticons, and labeled them with positive or negative sentiment; e.g., :) and :D express positive sentiment, whereas :(and :-(express negative sentiment. Positive or negative sentiment is then assigned to every message containing emoticons. Alternatively, any reliable sentiment classifier, such as OpinionFinder [9], can be used to determine the sentiments. In our experiments, we choose emoticons to label messages, and then have $K = 2$ sentiment classes.

The inactive users who participate in less than 5 cascades, or very short cascades whose size is less than 8, are removed from the remaining cascades with sentimental messages. Finally, we have 6,219 active users, and 325 positive and 412 negative cascades of a total size 44,021 tuples.

[1]Sina Weibo (http://www.weibo.com).

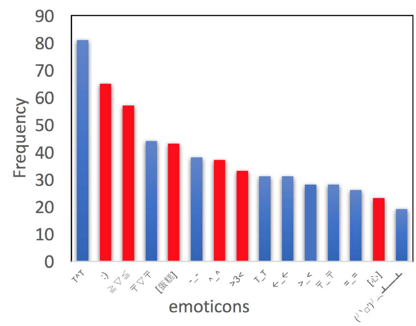


Fig. 3. The distribution of positive (red) and negative (blue) emoticons.

Microblog data WB2016: In the same way, we process the Microblog data sampled from Sina Weibo database, during a week from June 1, 2016. And finally we have totally 5,601 active users, and 457 positive and 130 negative cascades of total size 36,465 tuples for evaluation.

Figure 3 illustrates the distributions of top frequently used emoticons in the messages of cascades of WB2013, indicating their positive sentiments or negative sentiments.

4.2 Experimental Setup

In the experiments, we chose the following models for comparisons:

— *CT Bern* and *CT Jacc* models [26]: CT Bernoulli and CT Jaccard that are continuous time (CT) models with the propagation probability P_{ab} from infected user a to user b decays over time. For a fair comparison, we used the same decaying function as our model:

$$P_{ab} = P^0_{ab}/(t_b - t_a + 1).$$

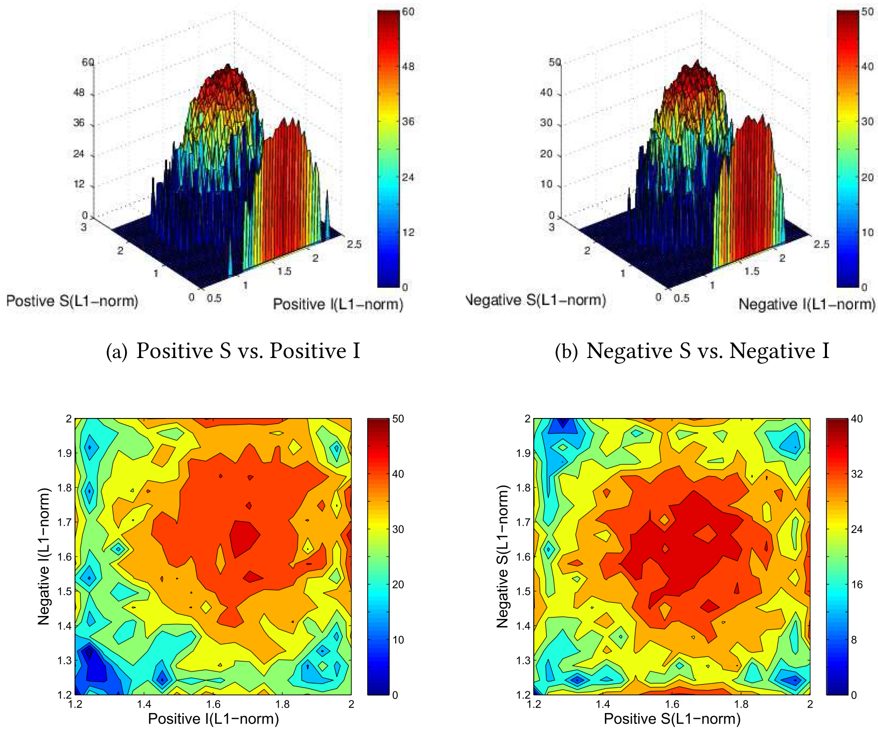
The CT Bernoulli model assumes that an initial propagation probability $P^0_{\cdot}$ follows Bernoulli distribution, i.e., the number of instances of successfully passed information divided by the total number of trials. The CT Jaccard model defines an initial propagation probability $P^0_{\cdot}$ of a user pair in a form of Jaccard index. The probability is calculated as the number of instances of successfully passed information divided by the total number of cascades containing at least one of the users in a pair. Since the causality relationships are not available in training data, the successful passes take place from every earlier users to the latter one in a cascade.

— *NetRate*[21]: This model directly defines a scalar parameter as an interpersonal influence between a pair of users, and it learns them by using the Survival Analysis Model. We use the Jaccard Index to initialize the parameters to get better fine-tuned results.

— *CT LIS*: Here, we ignore the differences between latent influence representations and susceptibility representations on sentimental polarities in our model. We set $K = 1$ and $o = 1$. Since the model is a CT model, we add CT into the name.

— *Sent LIS*: This is a model that learns influences on sentiments by considering all the negative cases.We use "Sent LIS (ns)" to indicate when our model uses negative sampling.

The complete dataset was split into 10 groups for cross-testing. Thus, each experiment was repeated 10 times, and the average results and the standard deviation (Std) were calculated. For computational efficiency, the dimension D of users' representations on a sentimental polarity is set at 8.



(a) Positive S vs. Positive I

(b) Negative S vs. Negative I

(c) Negative I vs. Positive I

(d) Negative S vs. Positive S

Fig. 4. Analysis of L1-norm of latent influence representations and susceptibility representations on sentimental polarities.

4.3 Study of Users' Influences and Susceptibilities

In this section, we present the different distributions of influential and susceptible users, and interesting patterns in their behaviors. Our Sent LIS model learned the representations of matrix $\mathbf{I}_v$ and matrix $\mathbf{S}_v$ for user v. Each row of $\mathbf{I}_v$ and $\mathbf{S}_v$, respectively, gives the representation vectors of user v's influence and susceptibility on the corresponding sentiment polarity. With sentiments of positive and negative, a user has four types of vectors: Positive I, Negative I, Positive S, and Negative S. We study those row vectors with L1-norms, i.e., the summation of the absolute value of each element. Thus, we can use those L1-norms as coordinates of a point for each user. Figure 4 shows the contour maps to visualize the user distributions in different influence and susceptibility coordinate systems.

The visual representations of "Positive S vs. Positive I" and "Negative S vs. Negative I" (Figure 4(a) and (b)) both interestingly show two groups of users in each sentiment class, corresponding to two peaks in each figure. The group along the bottom axis has very lower susceptibilities but significant influences. The upper right group has both significant influences and susceptibilities. We will refer to these two groups of people as *primary influential* and *secondary influential*, respectively. The primary influential people have the power to influence other users, without being infected easily, as was also explained by Aral and Walker on Facebook users [1]. Interestingly, on Sina Weibo, users can get influence credit, or have more people retweet them, by being an agency or hub. So people invest considerable effort in retweeting (potentially) interesting messages to attract others to follow them and retweet from them. This type of user is not restricted to the Sina Weibo scenario. Some primary influential people like to explore new restaurants and make recommendations to their friends or audiences. Another type of user simply likes to browse other people's recommendations or ratings and advertise them to their own audiences. Similar activities might be seen in POI recommendation, hotel bookings, and online shopping as well. As for an online system, both primary influential and secondary influential people are important. The primary influential users represent the originating power of the system to bring new resources and initiate a cold start. The secondary influential users are good advertisers or hubs to enable other people to get information efficiently. As the primary influential people are considered high-quality users by system operators, the secondary influential ones may easily be persuaded to help in sharing advertisements.

Figure 4(c) and (d) shows that users may have different influences on positive and negative emotions or may be more susceptible to different sentimental messages. In the figure showing "Negative I vs. Positive I," users could have higher influences on positive sentiment, but lower influences on negative sentiment, and vice versa, as the "heat" spreads out over a wide area. A group of users has almost the same significant influences on both sentimental polarities, as the hottest spot shows. Similarly, in the figure showing "Negative S vs. Positive S," some users are more sensitive to the messages with positive emotion whereas other users are more susceptible to negative emotion. A larger group of users has the same significant susceptibilities, as a larger hot area is located in the middle of the figure.

4.4 Predicting Cascade Dynamics

The application of predicting cascade dynamics (PCD) aims at predicting whether a user v will take an action (i.e., become infected) given time t. It is useful to find out who is probably the next buyer of recommendations before time t. There are two ways to measure the prediction. One is to use a metric for binary classification. The infected users belong to the positive class, and the survivors belong to the negative class. For every user, we calculate the likelihood of becoming infected.

Thus, with the likelihood values for all the users, a true positive (TP) rate and a false positive (FP) rate can be calculated given any threshold. The ROC curve is then drawn with the TP rate and FP rate as the coordinates. After that, the classification metric AUC (the area under the ROC curve) is evaluated following [16]. The other way of measuring the prediction is by using a ranking metric. Since we have the likelihood of candidate users becoming infected in a given time, the users of highly ranked likelihood will probably be infected. A higher-performing model is expected to give a higher rank to the user in the ground truth. Therefore, we use a ranking metric, Mean Reciprocal Rank (MRR) [51]:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i},$$

where $|Q|$ is number of tests of predicting next activated user, and $rank_i$ is the rank of corrected user in the ith prediction tests.

Experimental results. We first validate our algorithm on learning users' influence and susceptibility vectors by comparing the performance of PCD task on our synthetic data. Since sentiment classes are modeled as independent dimensions, we set $K = 1$ for considering influence and susceptibility on any class. Table 1 illustrates the results our algorithms and the baselines. As we can see, our algorithms achieve better metrics, i.e., higher AUC and MRR, than the baselines, which show the effectiveness of our algorithms with/without negative sampling techniques.

Table 1. Evaluations on Synthetic Data

		CT Bern	CT Jacc	NetRate (Jacc)	CT LIS	CT LIS (ns)
AUC	Avg	0.4558	0.4344	0.3027	**0.5096**	**0.6501**
	Std	±0.0548	±0.0769	±0.0523	±0.0934	±0.0916
MRR	Avg	0.0032	0.0028	0.0047	**0.0228**	**0.0206**
	Std	±0.0007	±0.0006	±0.0023	±0.0051	±0.0039

Both the best and the second best results are bold.

Table 2. Evaluations on PCD Task for Two Real-World Datasets

			CT Bern	CT Jacc	NetRate (Jacc)	CT LIS	Sent LIS	Sent LIS (ns)
WB2013	AUC	Avg	0.8732	0.8621	0.8718	0.8793	**0.8992**	**0.8983**
		Std	±0.0658	±0.0802	±0.0730	±0.0207	±0.0152	±0.0156
	MRR	Avg	0.0062	0.0064	0.0071	0.0196	**0.0216**	**0.0265**
		Std	±0.0029	±0.0036	±0.0038	±0.0039	±0.0033	±0.0044
WB2016	AUC	Avg	0.9273	0.9250	0.9253	**0.9339**	**0.9334**	0.9242
		Std	±0.0222	±0.0210	±0.0201	±0.0202	±0.0221	±0.0212
	MRR	Avg	0.0138	0.0195	0.0202	0.0052	**0.0218**	**0.0292**
		Std	±0.0064	±0.0127	±0.0125	±0.0012	±0.0056	±0.0037

Both the best and the second best results are bold.

In terms of real-world data, the results of cross tests are shown in Table 2 for both WB2013 and WB2016 datasets. The average (Avg) and Std are reported for metrics, and the best and the second best averages are shown in bold text. We can see that our models in both classification and ranking metrics perform the best, when compared with the list-wise models. Our models with sentiments achieve nearly 0.9 in AUC, better than the baselines (less than 0.87) in WB2013. CT LIS and Sent LIS perform the best in WB2016, while the former without sentiments performs a little better, which may be the cause of unbalanced sentiment labels in the data. In terms of ranking metric MRR, the sentimental models, i.e., Sent LIS and Sent LIS (ns) outperform the others in both WB2013 and WB2016. Moreover, the machine learning model NetRate further fine-tunes the Jaccard index to get better MRR and AUC. Above all, both the classification metric and the ranking metric show that our models perform better in cascade dynamics prediction than the pair-wise CT Bernoulli, CT Jaccard, and NetRate. Such experimental results demonstrate the advantages of our models in the consideration of influence dependency and the reduction of overfitting by reducing model complexity. Furthermore, considering sentiments can improve the performance of learned representations of users' influences.

The ROC curves in Figure 5 show that our models achieve better performance than list-wise models, considering the areas under the curves. All our variants, i.e., CT LIS, Sent LIS, and Sent LIS (ns) achieve higher AUC values in the classification metric. With Jaccard index as initialization, NetRate improves the performance of the CT Jaccard model.

4.5 Cascade Size Prediction

CSP is a key part of influence maximization and viral marketing applications. CSP helps people to know if a message, web, product, or place will eventually become popular. In the application, we chose the first P tuples in each cascade as the initial status, and we predicted the cascade size at time $t_N, t_N > t_P$. t_P is the infecting time of the last user in the initial tuples. We made our prediction by simulating the generation process of a cascade. First, the time interval $t_N - t_P$ is evenly split

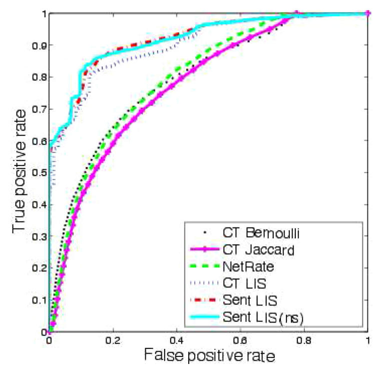


Fig. 5. Our models output the best ROC curve on prediction of next infected user.

Table 3. Evaluations on CSP Task for Two Real-World Datasets

			CT Bern	CT Jacc	NetRate (Jacc)	CT LIS	Sent LIS	Sent LIS (ns)
WB2013	MAPE	Avg	0.7199	0.7105	0.7109	**0.6259**	**0.6259**	**0.6362**
		Std	±0.0270	±0.0333	±0.0350	±0.0883	0.1458	±0.0225
WB2016	MAPE	Avg	0.8897	0.6048	0.5849	**0.5210**	**0.5313**	0.5656
		Std	±0.2699	±0.1164	±0.1313	±0.0792	±0.0879	±0.0888

Both the best and the second best results are bold.

into discrete time points. Thus, at the first time scale τ_{P+1} right after t_P, each infected user u tries to influence an uninfected user v with the probability $Pr(T \leq \tau_{P+1}|t_u; \phi(\mathcal{H}_{u,v}))$. Starting from the time point right after t_P, every infected user u makes a random trial to infect user v at each time point $\tau_i > t_P$. The probability in a trial is

$$Pr(T \leq \tau_i|t_u; \phi(\mathcal{H}_{u,v})) = \frac{\int_{\tau_{i-1}}^{\tau_i} f(t|t_u; \phi(\mathcal{H}_{u,v}))dt}{S(\tau_{i-1}|t_u; \phi(\mathcal{H}_{u,v}))}.$$

This is a conditional probability, given that user v will survive until time τ_{i-1}. Users who become infected in a random trial are added to the number of infected users for the next step of the simulation process. Thus, we use the *mean absolute percentage error* (MAPE) to measure the predictions with ground truth cascade sizes, where a smaller value indicates a better prediction.

Experimental results. We chose first $P = 10$ users in a cascade as the initialization for the prediction. The simulations for prediction were repeated 100 times for every testing cascades. We averaged the results at the end of prediction time in Table 3. We can see that the best CSP results all come from our models in both datasets. In WB2013, our best model achieves at least 11.9% MAPE reduction, compared to the baseline models. And our best model also reduces 10.9% MAPE value, compared to the best baseline (NetRate) in WB2016. Thus, our learned influences and susceptibilities from the proposed models are effective to improve the well-known CSP problem.

4.6 Who will be Retweeted

"Who will be retweeted" (WBR) is a new application that helps to identify the causality of who influences whom to take an action. For example, it enables us to find out who has influenced or motivated the target user to retweet a message, "like" a webpage, buy a product, or go to some location. Sina Weibo supports users to retweet a message from a retweeter instead of the original

Table 4. Evaluations on “Who will be Retweeted” Task

		CT Bern	CT Jacc	NetRate (Jacc)	CT LIS	Sent LIS	Sent LIS (ns)
Acc	Avg	0.1221	0.3000	0.3005	**0.4123**	0.3840	**0.3980**
	Std	±0.0365	±0.0964	±0.0961	±0.0874	±0.1255	±0.1392
MRR	Avg	0.2592	0.4349	0.4354	0.4696	**0.4822**	**0.4920**
	Std	±0.0703	±0.1275	±0.1273	±0.0876	±0.1269	±0.1348

Both best and second best results are bold.

author. Meanwhile Sina Weibo automatically generates marks to indicate who retweets whom. By extracting the relationships, one can identify the process of who influences whom, which is treated as the ground truth for WBR tasks.

Our model assumes that all the previously infected users have a chance to influence or infect the current user, so that there can be multiple exposures, only one of which can be successful. In the multiple exposures, every infected user has an interpersonal likelihood of infecting the target user. Thus, the user with higher interpersonal likelihood has a greater probability of being retweeted by the current user as the causality. Given (v_i, t_i), namely, user v_i infected at time t_i, the user that v_i retweets is

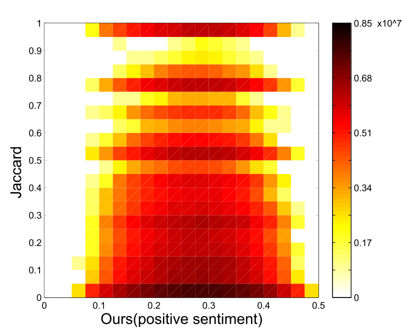
$$arg \max_{j:t_j<t_i} f(t_i|t_j; \phi(\mathcal{H}_{ji})).$$

Therefore, we use two ranking metrics for this task: average *Accuracy* (Acc) of top-one prediction, and MRR. Larger values of Acc and MRR indicate better performances.

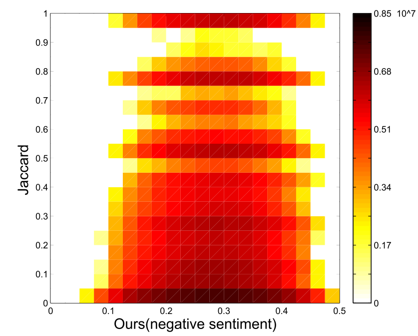
Experimental results. Our results in causality judgment of who influences whom achieve more than 37.2% and 13.0% increases in the accuracy metric and MRR metric, respectively, as Table 4 shows. WBR task is only evaluated on WB2013, since retweets marks are not available in WB2016. In the table, the top-one accuracy (Acc) and MRR are averaged for the 10-fold cross tests as well. CT LIS, Sent LIS, and Sent LIS (ns) outperform the pair-wise models, without suffering from the overfitting problem in causality judgment. Compared with NetRate, our models improve 37.2%, 27.8%, and 32.4%, respectively, in accuracy at top-one, and improve 7.9%, 10.7%, and 13.0%, respectively, in MRR. With negative sampling, Sent LIS can balance the positive and negative cases on one hand, and on the other hand it can consider the information of negative cases in an expectation, resulting in a better choice of descent gradient. In turn, it achieves better performances in both accuracy and MRR than Sent LIS without sampling. In addition, in the comparisons among pair-wise models, CT Jaccard still holds an advantage over CT Bernoulli in both metrics. NetRate (Jaccard) is the best of the three, due to the machine learning with the Survival Analysis Model.

4.7 Discrimination of Interpersonal Influences

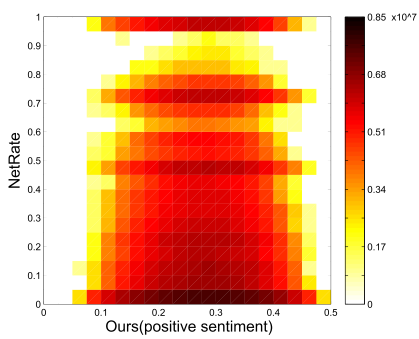
With the analysis of estimated interpersonal influences, Figure 6 shows that our values are more discriminating, guaranteeing a better application performance. We separately calculated the interpersonal influences using Equation (2) on positive sentiment and negative sentiment, for each user pair. We use the propagation probability output by CT Jaccard model as interpersonal influences. In Figure 6(a) and (b), we construct a point with the interpersonal influence of Sent LIS (positive or negative) as an X-coordinate, and interpersonal influence of Jaccard as a Y-coordinate, for each pair of users. All the points are put into lattices, and a heat map is used to color each lattice cell based on the point count. In the same way, NetRate uses normalized transmission rates as the estimation of interpersonal influences. We draw the heat maps for our Sent LIS vs. NetRate in Figure 6(c) and (d). In the four sub-figures, very hot and dark cells lines appear on the X-axis from 0.1 to 0.4. Such



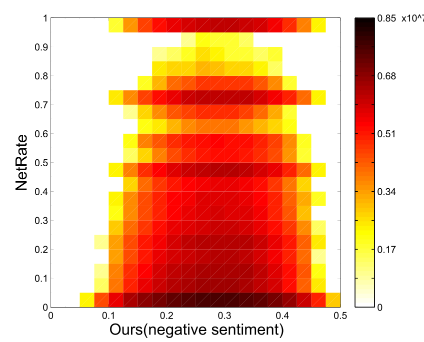
(a) Ours vs. CT Jaccard's on positive sentiment



(b) Ours vs. CT Jaccard's on negative sentiment



(c) Ours vs. NetRate's on positive sentiment



(d) Ours vs. NetRate's on negative sentiment

Fig. 6. Analysis of interpersonal influences.

lined cells show very small influence values for both the CT Jaccard and NetRate models, resulting from the overfitting. For the influences of those user pairs, our model discriminates the influential degrees from 0.1 to 0.4. In addition, our model shows a discriminating distribution of interpersonal influences at the higher values of the evaluation models. Therefore, all the above information gives evidence that our learned influences are more discriminating, offering good performances in the above applications.

5 CONCLUSIONS

In this study, we have proposed a model to learn the distributed representations of users' influences from their historical behaviors. By explicitly characterizing the sentimental influence and susceptibility of each user with a pair of matrices, respectively, the model reduces the complexity of pair-wise models, and considers influence dependencies. We also designed an effective algorithm with negative sampling to train the model, so as to balance the positive and negative cases while reducing the computational cost. Experiments on synthetic data validate the effectiveness of our learning algorithms for latent influence and susceptibility vectors.

Our model does not require the knowledge of social network structure, hence having wide applicability to the scenarios with or without explicit social networks. Explicit social network can be added as indicators in the likelihood of a user getting infected by the connected and effective ones. The evaluations on tree tasks and two real-world datasets show that our models are the best in most cases.

The advantages of our model are as follows:

— *Dependency awareness:* CT LIS can capture the dependency between two interpersonal influences, when being sent from the same user, or when being received by the same user.
— *Conciseness:* our model uses fewer parameters with an advantage in modeling sparse data.
— *Effectiveness:* our model achieved the best performance on two real-world Microblog datasets from 2013 and 2015, by evaluating on the three prediction tasks.
— *Unification of temporal and categorical information:* our model proposes a unified framework based on temporal model, to make use of infection time, and categorical attributes of sentiments.

Moreover, we analyzed two groups of people, called the primary influential and secondary influential groups. The former group creates initial and high-quality messages most of the time, to attract other users to retweet; the latter group attracts other people's attention by retweeting interesting messages already in the system. On the analysis of our estimated interpersonal influences, the results show discriminating values, which guarantees better application performance with the learned representations of users' influences.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Yongqing Wang for providing helps in some coding problems, and Mr. Bruce Barron for writing advice.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] S. Aral and D. Walker. 2012. Identifying influential and susceptible members of social networks. *Science* 337, 6092 (2012), 337–341.
[2] Yoav Artzi, Patrick Pantel, and Michael Gamon. 2012. Predicting responses to microblog posts. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*. 602–606. Retrieved from http://dl.acm.org/citation.cfm?id=2382029.2382126.
[3] Younggue Bae and Hongchul Lee. 2012. Sentiment analysis of Twitter audiences: Measuring the positive or negative influence of popular twitterers. *Journal of the American Society for Information Science and Technology* 63, 12 (2012), 2521–2535.
[4] Peng Bao, Hua-Wei Shen, Xiaolong Jin, and Xue-Qi Cheng. 2015. Modeling and predicting popularity dynamics of microblogs using self-excited Hawkes processes. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 9–10.
[5] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.
[6] David Bindel, Jon Kleinberg, and Sigal Oren. 2015. How bad is forming your own opinion? *Games and Economic Behavior* 92 (2015), 248–265.
[7] Zhenpeng Chen, Sheng Shen, Ziniu Hu, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. 2019. Emoji-powered representation learning for cross-lingual sentiment classification. In *the World Wide Web Conference*. ACM, 251–262.
[8] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. 2013. StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*. 509–518.
[9] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. 355–362.
[10] R. Crane and D. Sornette. 2008. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences* 105, 41 (2008), 15649–15653.
[11] Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya, and Manuel Gomez Rodriguez. 2016. Learning and forecasting opinion dynamics in social networks. In *Advances in Neural Information Processing Systems*. 397–405.
[12] Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. 2000. Mixing beliefs among interacting agents. *Advances in Complex Systems* 3, 01n04 (2000), 87–98.

[13] Morris H. DeGroot. 1974. Reaching a consensus. *Journal of the American Statistical Association* 69, 345 (1974), 118–121.

[14] Nan Du, Yingyu Liang, Maria-Florina Balcan, Manuel Gomez-Rodriguez, Hongyuan Zha, and Le Song. 2016. Estimating diffusion networks: Recovery conditions, sample complexity & soft-thresholding algorithm. *Journal of Machine Learning Research* 17, 90 (2016), 1–29.

[15] Nan Du, Le Song, Hyenkyun Woo, and Hongyuan Zha. 2013. Uncover topic-sensitive information diffusion networks. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*. 229–237.

[16] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874.

[17] Santo Fortunato, Vito Latora, Alessandro Pluchino, and Andrea Rapisarda. 2005. Vector opinion dynamics in a bounded confidence consensus model. *International Journal of Modern Physics C* 16, 10 (2005), 1535–1551.

[18] Jinhua Gao, Huawei Shen, Shenghua Liu, and Xueqi Cheng. 2016. Modeling and predicting retweeting dynamics via a mixture process. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 33–34.

[19] Aristides Gionis, Evimaria Terzi, and Panayiotis Tsaparas. 2013. Opinion maximization in social networks. In *Proceedings of the SDM*. SIAM, 387–395.

[20] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1, 12 (2009).

[21] Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. 2011. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning*. 561–568.

[22] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1019–1028.

[23] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. 2012. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data* 5, 4 (2012), 21.

[24] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. 2013. Modeling information propagation with survival theory. In *Proceedings of the 30th ICML*. 666–674.

[25] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. 2013. Structure and dynamics of information pathways in online media. In *Proceedings of the 6th WSDM*. 23–32.

[26] A. Goyal, F. Bonchi, and L. V. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the 3rd WSDM*. 241–250.

[27] Rainer Hegselmann and Ulrich Krause. 2002. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation* 5, 3 (2002).

[28] Magnus R. Hestenes. 1969. Multiplier and gradient methods. *Journal of Optimization Theory and Applications* 4, 5 (1969), 303–320.

[29] Bryan Hooi, Kijung Shin, Shenghua Liu, and Christos Faloutsos. 2019. SMF: Drift-aware matrix factorization with seasonal patterns. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 621–629.

[30] Elihu Katz and Paul Felix Lazarsfeld. 1955. *Personal Influence, The Part Played by People in the Flow of Mass Communications*. Transaction Publishers.

[31] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

[32] Adam D. I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. 2014. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences* 111, 24 (2014), 8788–8790.

[33] Jerald F. Lawless. 2011. *Statistical Models and Methods for Lifetime Data*, Vol. 362. John Wiley & Sons.

[34] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1 (May 2007), Article 5.

[35] C. J. Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19, 10 (2007), 2756–2779.

[36] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. 2010. Mining topic-level influence in heterogeneous networks. In *19th ACM International Conference on Information and Knowledge Management.*199–208.

[37] Shenghua Liu, Xueqi Cheng, Fuxin Li, and Fangtao Li. 2014. TASC: Topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering* 27, 6 (2014), 1696–1709.

[38] Shenghua Liu, Houdong Zheng, Huawei Shen, Xueqi Cheng, and Xiangwen Liao. 2017. Learning concise representations of users' influences through online behaviors. In *Proceedings of the IJCAI*. 2351–2357.

[39] Wei-Xue Lu, Chuan Zhou, and Jia Wu. 2016. Big social network influence maximization via recursively estimating influence spread. *Knowledge-Based Systems* 113 (2016), 143–154.

[40] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez Rodriguez. 2015. Hierarchical Dirichlet Hawkes process for modeling the dynamics of online learning activity. In *Proceedings of theWorkshop on Networks in the Social and Information Sciences*.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems*. 3111–3119.

[42] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 61–70.

[43] Francisco A. Rodrigues and L. da F. Costa. 2005. Surviving opinions in Sznajd models on complex networks. *International Journal of Modern Physics C* 16, 11 (2005), 1785–1792.

[44] K. Saito, R. Nakano, and M. Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 67–75.

[45] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced Poisson processes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*. 291–297.

[46] Geoffrey M. Stephenson and Geoffrey T. Fielding. 1971. An experimental study of the contagion of leaving behavior in small gatherings. *Journal of Social Psychology* 84, 1 (1971), 81–91.

[47] Krzysztof Suchecki, Víctor M. Eguíluz, and Maxi San Miguel. 2005. Voter model dynamics in complex networks: Role of dimensionality, disorder, and degree distribution. *Physical Review E* 72, 3 (2005), 036132.

[48] Katarzyna Sznajd-Weron and Jozef Sznajd. 2000. Opinion evolution in closed community. *International Journal of Modern Physics C* 11, 06 (2000), 1157–1165.

[49] J. Tang, J. Sun, C. Wang, and Z. Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 807–816.

[50] Yu-Song Tu, AO Sousa, Ling-Jiang Kong, and Mu-Ren Liu. 2005. Sznajd model with synchronous updating on complex networks. *International Journal of Modern Physics C* 16, 07 (2005), 1149–1161.

[51] Ellen M. Voorhees. 1999. The TREC8 question answering track report. In *Proceedings of Text REtrieval Conference*.

[52] Yongqing Wang, Shenghua Liu, Huawei Shen, Jinhua Gao, and Xueqi Cheng. 2017. Marked temporal dynamics modeling based on recurrent neural network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 786–798.

[53] Yongqing Wang, Huawei Shen, Shenghua Liu, and Xueqi Cheng. 2015. Learning user-specific latent influence and susceptibility from information cascades. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

[54] Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. 2017. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*. 2985–2991.

[55] Yongqing Wang, Hua-Wei Shen, Shenghua Liu, and Xue-Qi Cheng. 2013. Learning user-specific latent influence and susceptibility from information cascades. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI, 477–483.

[56] Yichen Wang, Evangelos Theodorou, Apurv Verma, and Le Song. 2016. A stochastic differential equation framework for guiding online user activities in closed loop. In *Artificial Intelligence and Statistics (AISTATS'18)*. 1077–1086.

[57] Yichen Wang, Bo Xie, Nan Du, and Le Song. 2016. Isotonic Hawkes processes. In *Proceedings of the 33rd International Conference on Machine Learning*. 2226–2234.

[58] Duncan J. Watts and Peter Sheridan Dodds. 2007. Influentials, networks, and public opinion formation. *Journal of Consumer Research* 34, 4 (2007), 441–458.

[59] Brian E. Weeks, Alberto Ardèvol-Abreu, and Homero Gil de Zúñiga. 2017. Online influence? Social media use, opinion leadership, and political persuasion. *International Journal of Public Opinion Research* 29, 2 (2017), 214–239.

[60] Reza Zafarani, William D. Cole, and Huan Liu. 2010. Sentiment propagation in social networks: A case study in livejournal. In *Advances in Social Computing*. Springer, 413–420.

[61] Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701* (2012).

[62] Jie Zhang, Jiaqi Ma, and Jie Tang. 2016. Learning cascaded influence under partial monitoring. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'16)*. IEEE, 255–262.

[63] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1513–1522.

Received March 2018; revised May 2019; accepted August 2019