Learning Influence Probabilities and Modelling Influence Diffusion in Twitter

Zizhu Zhang, Weiliang Zhao and Jian Yang
Macquarie University
Sydney, Australia
zizhu.zhang@hdr.mq.edu.au,{weiliang.zhao,jian.yang}@mq.edu.au

Cecile Paris and Surya Nepal
CSIRO
Sydney, Australia
{cecile.paris,surya.nepal}@data61.csiro.au

ABSTRACT

Influence diffusion has been widely studied in social networks for applications such as service promotion and marketing. There are two challenging issues here: (1) how we measure people's influence on others; (2) how we predict whom would be influenced by a particular person and when people would be influenced. Existing works have not captured the temporal and structural characteristics of influence diffusion in Twitter. In this paper, we firstly develop a model to learn influence probabilities between users in Twitter from their action history; secondly, we introduce diffusion models that are used to predict how information is propagated in Twitter. Experiment results show that our proposed models outperform existing models in terms of the balanced precision and recall.

KEYWORDS

influence probability; influence diffusion; time decay; depth decay; Twitter

ACM Reference Format:

Zizhu Zhang, Weiliang Zhao and Jian Yang and Cecile Paris and Surya Nepal. 2019. Learning Influence Probabilities and Modelling Influence Diffusion in Twitter. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW'19 Companion), May 13–17, 2019, San Francisco, CA, USA*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316701

1 INTRODUCTION

People's influence in social media can be measured by others' reactions to their postings [16]. There has been increasing interest in influence diffusion in social networks due to its applications in viral marketing [11], protest event detection [1], rumour control [6], etc. Two interesting and yet challenging issues arise: (1) how we measure people's influence on others; and (2) how we predict whom would be influenced by a particular person and when people would be influenced.

Following Kempe et al.'s seminal work in 2003 [11], simple heuristic influence probability assignments (e.g., uniform models and degree-weighted models) are being used [5]. These works are not concerned with influence probability calculation between two users. A few studies [10, 12, 15] have developed methods in learning influence probabilities from users' historical action data in Twitter. Designed for information sharing, Twitter is an ideal social network platform to investigate influence diffusion due to its *explicit interaction* mechanisms such as "retweet", "like", "reply" and "quote". For example, we can say intuitively that user v is influenced by user u if v retweets u's message [9, 12]. Twitter users can also be influenced implicitly via the *follow* mechanism, with which users are exposed to the messages posted by whom they follow. As a matter of fact there is a high possibility that a user posts a new message without any explicit interactions with other messages (a message we call it a "tweet") based on the messages of whom she or he follows. We refer to this case as an *implicit interaction*. An example is given in Table 1: user $u3391$ posts "Marriage Bill has passed", and 139 minutes later, $u849$, who follows $u3391$, posts a "tweet" of similar content. Existing works mainly use some of the explicit interactions to get the estimated influence probabilities between users. The "like" and implicit interactions are largely ignored. In our *Darwin* dataset (to be introduced in Section 6.1), the number of "likes" is larger than the sum of all the other explicit interactions, and 64% more interactions can be captured if implicit interactions are considered. It may lead to inaccurate estimations by ignoring them. Covering all of v's reactions to u's messages, explicitly and implicitly, is crucial to accurately measure u's influence on v in Twitter.

Now we address the second issue: how do we predict whom will be influenced by a particular person at what time? To answer this question, we need to look at how influence diffuses. In Twitter, the influence can change with time and depth when going through the network. The well-known diffusion models in [11] use unchanged influence probabilities, which are not suitable for Twitter.

In order to overcome the limitations of the existing works mentioned above, we propose an approach for influence probability calculation and influence diffusion modelling in Twitter. We collect two Twitter datasets, referred to as *Darwin* and *MelCup17*, to evaluate the performance of proposed models. The contributions of this paper are summarised as follows.

- To learn the pair-wise influence probabilities in Twitter, we utilise all the available explicit interactions, including the ignored "like" in existing works, and study the implicit interactions via the *follow* mechanism;
- Time and depth-sensitive diffusion models are proposed for Twitter to predict whom would be influenced by a particular user and when people would be influenced;
- Experimental results against two real-world datasets show that our proposed diffusion models outperform existing ones in terms of the balanced precision and recall, especially to predict when people will be influenced.

The reminder of the paper is organised as follows. Section 2 reviews the related work. Section 3 provides the specifications of influence in Twitter. Section 4 presents our methods for influence

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316701

Table 1: Example of implicit influence

uid	time	text
$u3391$	t-139min	RT: The **Marriage Bill** has **passed** the #Senate
$u849$	t	Big news! Same-sex **marriage bill passes** Senate

$u849$ follows $u3391$.
$u849$'s tweet is posted 139 minutes later than $u3391$'s message.
The two messages have similar content.

probability calculation and influence diffusion modelling. We discuss the evaluation metrics in Section 5 . The experimental results are given and analysed in Section 6. We conclude in Section 7.

2 RELATED WORK

Existing work mainly use three kinds of heuristics to assign influence probabilities: uniform, trivalency and degree-weighted [5]. Saito et al. [19] are the first to focus on learning edge probabilities from diffusion logs. The probabilities for all the edges are iteratively computed using Expectation Maximization algorithm. Their method suffers the scalability due to that it updates the influence probability in each iteration. Goyal et al. [9] propose a method of learning influence probability from users' action logs sorted by action types and time. In the evaluation, they scan each actual action record and adopt an instance of General Threshold Model to update $p_v(S)$, which is the probability that v will take a particular action considering a set of his neighbours S having taken the action. Then $p_v(S)$ is compared with a randomly chosen threshold θ_v to do the prediction. While their way of evaluation is elegant, it is not able to simulate who will influence whom given just the user(s) who first take an action. Kutzkov et al. [12] learn $p(u, v)$ in Twitter data streams. They only study the "retweet probability" by adopting the ratio idea in [9]. Mei et al. [15] cover more explicit interactions, such as "reply", "mention" and "like", in the probability calculation. Jendoubi et al. [10] use belief functions to combine multiple aspects of influence indicators, including "retweet", "mention", and the "number of followers", to estimate influence in Twitter. None of these studies consider the implicit influence and fail to catch u's actions that have influenced v in a high degree of accuracy.

Another area related to this study is analysing influence diffusion in social networks. A number of well-known models are discussed in [11]. Modelling the spread of information and influence maximization have been studied in [1, 10, 15, 21]. Their diffusion models, assuming that the pair-wise influence probabilities are unchanged as the diffusion proceeds, can not capture the characteristics of influence diffusion in Twitter. Their solutions cannot be adopted to the Twitter environment for influence prediction. An early work of analysing patterns of cascading behavior in blog data [13] reports that the most popular shapes of reposting cascades are wide and shallow trees, and the cascade sizes are mostly small. There are similar observations on URL cascades [2, 7] and retweet cascades [20] in Twitter: the majority of messages (ranging from 73% to 95% across domains) in Twitter do not spread at all. This is because that "retweets" and "likes" cannot get any influence credit according to Twitter's mechanism design. In addition, time plays an important role in the diffusion of influence [17]. To predict influence diffusion in Twitter, the above features need to be properly considered. Our proposed diffusion models incorporate the temporal and structural features of influence cascades in Twitter.

3 INFLUENCE IN TWITTER

In this work, we discuss influence in Twitter according to users' actions. *Taking an action* in Twitter is *posting a message.* A message can be a "retweet", a "like", a "reply" or a "quote" (as explicit interactions with other messages); it can also be an independent "tweet" without any explicit relation to any other messages. As discussed earlier "tweets" can have implicit interactions with other messages via the *follow* mechanism. Our definition of influence in Twitter is:

Definition 1. Given an individual user u who posts a message, if a user v reacts to u's message, we define that u has influenced v. The following five kinds of reactions performed by v to u's message are regarded as influence:

- v retweets u's message
- v likes u's message
- v replies to u's message
- v quotes u's message
- v, who follows u, posts a new tweet that with similar contents to u's message within a certain time period (defined in Section 4.1.2).

The ***influence probability*** $p(u, v)$ is defined as the probability that v will react to u's message. $p(u, v)$ is learned from the action history, based on the ratio of the number of u's actions that have influenced v to the total number of u's actions. Based on the pairwise influence probabilities, we develop influence diffusion models to predict influence cascades [13] in the network, i.e., whom would be influenced by a particular person and when people would be influenced.

4 INFLUENCE PROBABILITY AND DIFFUSION

4.1 Learning Influence Probabilities

We construct the influence network $G_{Infl} = (V, E, P)$, where V is the set of users, E is the set of directed edges, and P is the set of influence probabilities between users, from both the follower network and users' action history. The action history, denoted as $\mathcal{A}$, is a set of all the messages posted by the users in V in a period. $\mathcal{A}_u$ denotes the set of all the messages of a particular user u, including u's "tweets", "retweets", "likes", "replies" , and "quotes", which are denoted as $\mathcal{A}_{u\_TWT}$, $\mathcal{A}_{u\_RET}$, $\mathcal{A}_{u\_LIK}$, $\mathcal{A}_{u\_REP}$, and $\mathcal{A}_{u\_QUO}$ respectively. $\mathcal{A}_u = \mathcal{A}_{u\_TWT} \bigcup \mathcal{A}_{u\_RET} \bigcup \mathcal{A}_{u\_LIK} \bigcup \mathcal{A}_{u\_REP} \bigcup \mathcal{A}_{u\_QUO}$. Also, let A_u denote a single message posted by u in $\mathcal{A}_u$.

4.1.1 The influence network structure. For any pair of $(u, v) \in V$, if there is a "follow" relation, or an explicit interaction from v to u, we add a directed edge (u, v) to G_{Infl} to indicate that u potentially influences v. For example, in Figure 1, user H follows B, and thus B has the potential to influence H, and in the derived G_{Infl} (the graph on the right), there is a directed edge (B, H); and H likes G's tweet, so a directed edge (G, H) is added to G_{Infl}.

4.1.2 Learning the implicit influence. As we discussed earlier, people can be influenced implicitly via the *follow* mechanism. The influence can be inferred from the content relations of messages

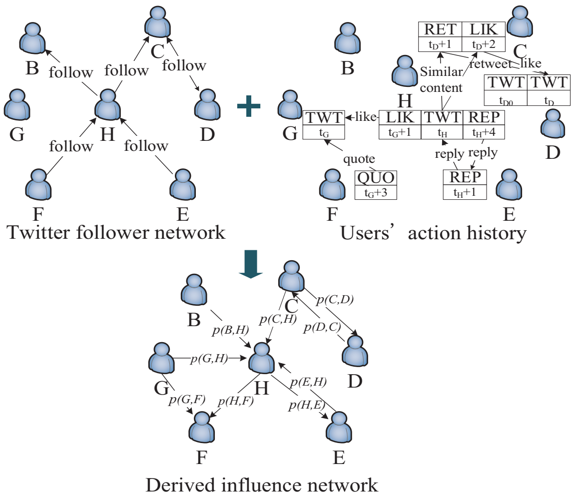


Figure 1: Constructing the influence network

posted by users. The challenge is to identify those tweets that are the results of following the people who post related messages. Tweets in $\mathcal{A}_{v\_TWT}$ are classified into two types: the tweets posted by user v without relations to messages of other users, denoted as $\mathcal{A}_{v\_TWT_{ini}}$; and the tweets as the results of the influence of the people who v follows, denoted as $\mathcal{A}_{v\_TWT_{inf}}$. Our approach of identifying an A_{v_TWT} as an $A_{v\_TWT_{inf}}$ considers the social network structure, time and content similarity.

Because of the speed of Twitter streams, messages become quickly dismissed. A concept of *influential time period* T_0 of a message is introduced. We assume that it is the period during which a message can have influence; after that it has no influence. We consider u to have an implicit influence on v, if v, who follows u, posts a new tweet A_{v_TWT} based on u's message A_u within T_0 after u posted the message. After getting all the possible $\langle A_u, A_{v\_TWT}\rangle$ pairs for each A_{v_TWT} under the *follow* and T_0 constraints, we use the content similarity to identify whether A_{v_TWT} is an $A_{v\_TWT_{inf}}$ based on A_u. To decide whether two messages have the similar content, we check if they have same tokens as user mentions (`@username`) [4], hashtags(`#hashtag`) [18] and urls(`bit.ly`) [2]. If two messages have at least one common token, we regard that their topics are similar and A_{v_TWT} is an $A_{v\_TWT_{inf}}$ due to the influence of A_u. If not, we look into the details of these two messages. We remove special tokens, stop words and words with less than 3 letters, and perform stemming and lemmatization on the words. After the pre-processing, the longest message has 34 important words for our data and on average a message contains 8.5 important words (the Standard Deviation is 4.9). Since a message in Twitter is normally short, sharing same important words is an indication that the two messages are about a similar topic. In the *Darwin* dataset collected based on users' location, 92.8% of the $\langle A_u, A_{v\_TWT}\rangle$ pairs evaluated by important words have no common words, and 6% have only one common word. We check the message pairs sharing more than two important words and identify whether their topics are similar. It is found that when two Twitter messages share three or more important words, their content are similar with a probability of above 90%. We consider A_{v_TWT} to be an $A_{v\_TWT_{inf}}$ if there exists an A_u that shares three or more important words in common with A_{v_TWT}. For an A_{v_TWT}, if there is no A_u satisfying the above three constraints, it is identified as an $A_{v\_TWT_{ini}}$.

For an $A_{v\_TWT_{inf}}$, there may have more than one A_u. We cannot really know which A_u has influenced v. Facing a similar case in tracking the diffusion of the urls in Twitter, Bakshy et al. [2] propose three choices: "first influence" which assigns full credit to the user who takes action first, "last influence" which assigns full credit to the user who takes action most recently, and "split influence" that splits credit equally among all possible A_u. They report that the qualitative findings are identical though the values varied slightly across the three assumptions. Here we adopt "last influence" and assign full credit to the most recent A_u.

4.1.3 Learning the influence probabilities. We first consider the total number of u's actions $|\mathcal{A}_u|$. Each A_u can influence others implicitly. However, we observe that not every A_u is recorded to have explicit influence. For example, in Figure 1, user G posts a "tweet" at time t_G, H likes it at t_G+1, and F sees the "tweet" from H's "like" and quotes it at t_G+3. F's action is recorded as a reaction to G's "tweet", not H's "like" of G's "tweet". The reactions to a "retweet" or "like" will be counted to the user who posts the root message. Only "tweet", "quote" and "reply" are recorded to have explicit influence on others. We refer to them as "Explicit Influenceable Actions"(EIA), denoted as $\mathcal{A}^*$: $\mathcal{A}^*_u = \mathcal{A}_{u\_TWT} \bigcup \mathcal{A}_{u\_REP} \bigcup \mathcal{A}_{u\_QUO}$. Both the number of total actions $|\mathcal{A}_u|$ and EIAs $|\mathcal{A}^*_u|$ are learned when scan the action history.

Then we consider the number of u's actions that have influenced v. Based on $\mathcal{A}^*_u$ and $|\mathcal{A}_u|$, we learn u's actions that have influenced v in explicit ways, denoted as $\mathcal{A}_{u2v\_e}$, and u's successful actions influencing v in implicit ways, denoted as $\mathcal{A}_{u2v\_i}$, separately in the scanning. In Twitter, v is able to react directly with u's same message multiple times in various ways. For example, in Figure 1, user C has two reactions (liking and retweeting) to D, and both of these two reactions are to D's "tweet" posted at t_D. Only one of D's actions has influenced C. $|\mathcal{A}_{u2v\_e}|$ or $|\mathcal{A}_{u2v\_i}|$ is the number of u's unique actions that have influenced v explicitly or implicitly. It requires just one scan of the action history to get $|\mathcal{A}^*_u|$, $|\mathcal{A}_u|$, $|\mathcal{A}_{u2v\_e}|$ and $|\mathcal{A}_{u2v\_i}|$.

We calculate $p(u,v)$ based on the ratio of the number of u's actions that have influenced v to the total number of u's actions specified in Section 3. A user u's explicit influence on v, denoted as $p^e_{u,v}$, and implicit influence on v, denoted as $p^i_{u,v}$, are calculated separately as:

$$p^e_{u,v} = f(|\mathcal{A}^*_u|) \cdot \frac{|\mathcal{A}_{u2v\_e}|}{|\mathcal{A}^*_u|} \tag{1}$$

$$p^i_{u,v} = f(|\mathcal{A}_u|) \cdot \frac{\beta|\mathcal{A}_{u2v\_i}|}{|\mathcal{A}_u|} \quad \beta \in (0,1) \tag{2}$$

In Equation 1 and 2, if the denominator $|\mathcal{A}^*_u|$ or $|\mathcal{A}_u|$ is small, the ratio can be high (e.g. $\frac{|\mathcal{A}_{u2v\_e}|}{|\mathcal{A}^*_u|} = \frac{1}{1} = 1$). A small denominator means insufficient observations of actions, and we employ $f(x) = 1/(1+e^{(-(x-1))})$ as a scaling factor (see Figure 2) to adjust its effect. $f(x)$ is 0.5, 0.73, and 0.88 when x is 1, 2 and 3 respectively; $f(x)$ is 0.95 when x 4 and approaches to 1 while x is bigger. In Equation 2, a control parameter β in the range $(0,1)$ is applied on the implicit

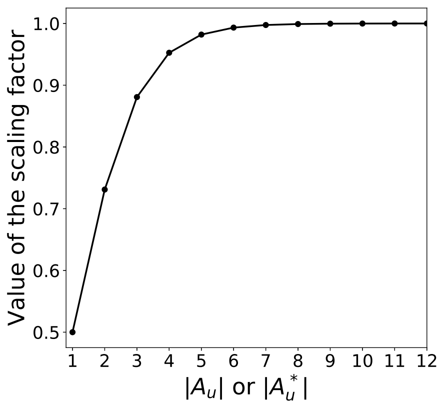


Figure 2: The scaling factor in Eq. 1 and 2

influence. We calculate p_2 with the joint probability of $p^e_{u,v}$ and $p^i_{u,v}$:

$$p_2 = 1 - (1 - p^e_{u,v})(1 - p^i_{u,v}) \tag{3}$$

It is assumed that there is always a small influence probability between connected user pairs in the influence network, which is denoted as p_1. $p(u, v)$ is calculated as:

$$p(u, v) = 1 - (1 - p_1)(1 - p_2) \tag{4}$$

$p(u, v)$ is time-related if T_0 constraint is applied to get $|\mathcal{A}_{u2v\_e}|$ and $|\mathcal{A}_{u2v\_i}|$. $p^{T_0}(u, v)$ is the probability that u's action will influence v in T_0's time since u takes the action; after T_0, the probability drops to 0. To set $|\mathcal{A}_{u2v\_i}|$ as 0, we get the probability that u influences v without considering the implicit influence.

4.2 Diffusion Models

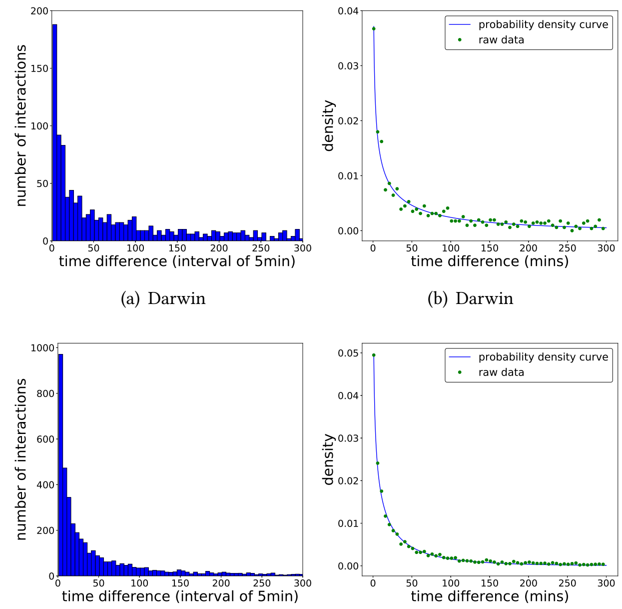
We apply the pair-wise influence probabilities to diffusion models to predict how influence diffuses among Twitter users. In the real world, the diffusion of influence is continuous, while cascade model proceeds in discrete steps from one time slot to the next. We assume each time slot is equal and choose the time slot t_s based on the temporal features of influence in learning data. After u takes an action, u is able to influence his/her out-degree neighbour v who has not been influenced regarding to this action in a maximum of $N = |\frac{T_0}{t_s}|$ steps, with a total probability of $p^{T_0}(u, v)$ and a probability $p_i(u, v)$ at the i^{th} step after u takes the action. We have

$$p^{T_0}(u, v) = 1 - \prod_i^N (1 - p_i(u, v)) \qquad (i = 1, 2, ..., N) \tag{5}$$

After N steps (i.e. a time period of T_0), u's influence on its out-degree neighbours drops to 0: $p_i(u, v) = 0 \quad (i > N)$. For an action, its influence cascade terminates if no more user is influenced.

For v who has not been influenced, denote S as the set of v's in-degree neighbours who are able to influence v. If there are multiple $u_j \in S$, we assume that the influence from multiple neighbours to v are independent of each other and adopt a joint probability $p_v(S)$, proposed by [9], to define the probability that v will be successfully influenced:

$$p_v(S) = 1 - \prod_{u_j \in S} (1 - p_i(u_j, v)) \tag{6}$$



(a) Darwin (b) Darwin

(c) MelCup17 (d) MelCup17

Figure 3: (a/b). Frequency of interactions vs. the time difference between interactions of *Darwin/MelCup17* (c). Fitting the frequency of interactions over time of *Darwin*

4.2.1 Time Constant Cascade Model(TC-C). In the Time Constant Cascade model, $p_i(u, v)$ is independent of time, which means at each time step i ($i \leq N$) after u takes the action, $p_i(u, v)$ stays the same.

$$p_i(u, v)_c = p_{i+1}(u, v)_c \qquad (i = 1, 2, ..., N - 1) \tag{7}$$

Then the constant $p_i(u, v)_c$ is solved from (5) and (7) as:

$$p_i(u, v)_c = 1 - (1 - p^{T_0}(u, v))^{1/N} \tag{8}$$

After i steps, the total probability that u influencing v is:

$$p^i(u, v)_c = 1 - \prod_1^i (1 - p_i(u, v)_c) \qquad (i = 1, 2, ..., N) \tag{9}$$

Because $p_i(u, v)_c = p_{i+1}(u, v)_c$, the joint influence probability from multiple neighbours when a new influential neighbour w adding in $p_v(S \bigcup w)$ and when a influential neighbour w becomes having no influence $p_v(S \setminus w)$ in TC-C models could be updated incrementally from $p_v(S)$ [9].

4.2.2 Time Decay Cascade Model(TD-C). We plot the number of explicit interactions against the time to show when v is influenced by u. The statistics of *Darwin* and *MelCup17* datasets are given in Figure 3(a) and 3(c) respectively. Each figure shows the frequency of interactions in intervals of 5 minutes over a span of 300 minutes. The majority of the interactions occur within a very short period of time and the number of interactions decay quickly. User interactions in Twitter are generally fast and short in lifespan [14]. We assume that for each (u, v) pair in a social network, $p(u, v)$ decays over time following the same fashion. To identify how $p(u, v)$ decays, we fit a probability density curve $p(x)$ based on the interaction frequency distribution over time, see Figure 3(b) and 3(d).

In the TD-C model, the relationships among $p_i(u, v)_d$ are as follows:

$$\frac{p_1(u,v)_d}{\int_0^{t_s} p(x)\,dx} = \ldots = \frac{p_i(u,v)_d}{\int_{(i-1)\cdot t_s}^{i\cdot t_s} p(x)\,dx} \qquad (i = 1, 2, \ldots, N) \quad (10)$$

After i steps, the total probability that u influencing v is:

$$p^i(u,v)_d = 1 - \prod_1^i (1 - p_i(u,v)_d) \qquad (i = 1, 2, \ldots, N) \quad (11)$$

With (5) and (10), the values of $p_i(u, v)_d$ can be obtained. Since $p_i(u, v)_d$ changes as the diffusion proceeds, the $p_v(S \bigcup w)$ and $p_v(S \setminus w)$ can not be computed incrementally.

For any $i \in [1, N-1]$, $p^i(u, v)_d$ in (11) is larger than $p^i(u, v)_c$ in (9). Due to limited space, we omit the proof here. It results that for the same $p^{T_0}(u, v)$, at any step before no more users can be influenced, the number of influenced users with the TD-C model will always be larger than that with the TC-C model.

4.2.3 Time-Depth Decay Cascade Model(TDD-C). Existing studies [7, 8] report that over 99% of the interactions to the root user in Twitter take place within one depth of the root user and the depth distribution of diffusion cascade graph is skewed. We consider the decay of influence with the influence diffusion cascade depth. Suppose that the shortest path length of an influenced user v to the root user u is $l(u, v)$, then in the TDD-C model the influence probability of v to its neighbour w, who has not been influenced at the i^{th} step since v takes the action, is calculated as

$$p_i(v,w)_{dd} = \delta^{l(u,v)} p_i(v,w)_d \quad (12)$$

δ is a decay factor for depth, ranging from 0 to 1. Only "reply" and "quote" to a root action can make the cascades go further and δ is learned as the ratio of the sum of "reply" and "quote" to the total number of explicit interactions.

4.2.4 Model Variations. With each of the three models, there are two variations depending on how $p(u, v)$ is learned. In total, there are six models: TC-Cp, TC-Cp_e, TD-Cp, TD-Cp_e, TDD-Cp, and TDD-Cp_e. p stands for the proposed models with probabilities learned from both explicit and implicit interactions, and p_e stands for the ones with probabilities learned from the explicit interactions only.

5 EVALUATION

TC-Cp_e has been studied in [9, 15] and is used as the baseline. Starting from an action initiated (i.e. a "tweet") by a root user u, the proposed diffusion models are used to simulate how this tweet is propagated through users in the influence network. With multiple rounds of simulations, the results give answers to the following questions: (a). On average, how many users can be influenced by u? (b). Who are the users that will be influenced by u? (c). When are these users influenced? (d). What are the shapes of the influence diffusion cascades like?

1. Predicting size The size of an influence diffusion cascade is the number of users involved in the cascade [13]. For a user u, $s(u)$ is the average number of users in the influence cascades of actions initiated by u according to the historical data, and $\hat{s}(u)$ is the predicted value. The difference between $s(u)$ and $\hat{s}(u)$ is represented with Normalise Mean Squared Error (NMSE):

$$NMSE = \frac{1}{n}\sum_i \left(\frac{s(u_i) - \hat{s}(u_i)}{s(u_i)}\right)^2$$

2. Predicting Shape For the influence cascade shape, we evaluate the depth distribution of the influenced users in the influence cascades graph. The depth of an influenced user is the shortest path length between this user and the root.

3. Predicting time The actual time v is influenced by an action is t_v and the $\hat{t}_v$ is the time that v is predicted to be influenced. In the applications of influence diffusion, such as viral marketing, the focus is how many users are influenced within a certain period of time [9]. If $\hat{t}_v \le t_v$, we consider that the model successfully predicts when v will be influenced.

4. Predicting who influences whom at what time For each A_{u_TWT} in the evaluation data, its actual influence is built up in the form of $(A_{u\_TWT}, u, v, v_{infled})$. v_{infled} marks whether v is influenced by A_{u_TWT}: it is 0 if v is not influenced, and if v is influenced, its value is the influenced time t_v. All users that have been influenced by A_{u_TWT} are denoted as $U_1(A_{u\_TWT})$. The out-degree neighbours of $U_1(A_{u\_TWT})$ and u in the influence network who are not influenced by A_{u_TWT} are the users whose $v_{infled} = 0)$, denoted as $U_0(A_{u\_TWT})$.

The simulated results of a simulated diffusion starting from an A_{u_TWT} are presented as a form of $(A_{u\_TWT}, u, \hat{v}, \hat{v}_{infled} = \hat{t}_v)$. Note that the diffusion models proceed in discrete steps from one time slot t_s to the next. A user is influenced at i^{th} ($i >= 1$) time slot, i.e. during a time period of $[(i-1)\cdot t_s, i\cdot t_s]$. We let $\hat{t}_v = (i-1)\cdot t_s$. The influenced users in the simulations are denoted as $\hat{U}_1(A_{u\_TWT})$.

For each action A_{u_TWT}, we get the TP, FN, FP, TN cases of predicting whom are influenced or not as follows: for each $v \in U_1(A_{u\_TWT}) \bigcup U_0(A_{u\_TWT})$,

- if $v_{infled} > 0$ and $v \in \hat{U}_1(A_{u\_TWT})$, it is a TP;
- if $v_{infled} > 0$ and $v \notin \hat{U}_1(A_{u\_TWT})$, it is a FN;
- if $v_{infled} == 0$ and $v \in \hat{U}_1(A_{u\_TWT})$, it is a FP;
- if $v_{infled} == 0$ and $v \notin \hat{U}_1(A_{u\_TWT})$, it is a TN.

By considering if a model predicts when v is influenced, the criteria of TP cases are more rigorous:

- if $v_{infled} > 0$ and $v \in \hat{U}_1(A_{u\_TWT})$ and $\hat{t}_v \le t_v$, it is a TP;
- if $v_{infled} > 0$ and $v \in \hat{U}_1(A_{u\_TWT})$ and $\hat{t}_v > t_v$, it is a FN.

In Twitter, $|U_1(A_{u\_TWT})|$ is relatively small compared to $|U_0(A_{u\_TWT})|$. As a result, the number of positive cases is much smaller than the number of negative cases in the actual data (TP, FN and FP cases will be much smaller than TN). The *precision*, *recall* and *F-score* measures are used to evaluate the prediction performance.

6 EXPERIMENTS

6.1 Datasets

This work learns the influence probabilities between users from their historical action data. Existing open datasets [6, 22] are not suitable for this purpose due to that they have not included rich

Table 2: Description of the Actions and Interactions

Datasets	action data				interaction data	
	#statuses	#likes	#actions	#users	#intractions	#users
Darwin-lrn	69,323	60,481	129,804	1,265	6,793	437
Darwin-eva	47,749	40,550	88,229	1,130	3,127	375
MelCup17-lrn	12,977	11,024	24,001	1,484	12,894	1,391
MelCup17-eva	10,567	9,266	19,833	1,374	7,670	1,132

Table 3: Description of the Influence Network

Dataset	following network		interaction network		influence network	
	# users	# edges	# users	# edges	# users	# edges
Darwin	3,880	58,503	437	2,080	3,883	58,799
MelCup17	1,484	70,756	1,391	7,403	1,484	72,710

interactions. We collect two datasets [1] from Twitter: one is based on users' location, and the other one is based on an event.

Location-based Dataset-"*Darwin*" We first identify 6,571 Twitter users whose location are "Darwin, Northern Territory, Australia" and collect their data real time from 16/11/2017 to 27/12/2017, Australian Central Standard Time(ACST) for learning (*Darwin-lrn*). The data includes "tweet", "retweet", "reply", and "quote", which are named as "statuses". There are 69,323 statuses, including 19,164 tweets. We then collect 60,481 "likes" of these users in this period. In total, we get 129,804 messages/actions posted/taken by 1,265 users. There are 6,793 (excluding self-interactions) interactions among 437 users, resulting an interaction network of 2,080 edges, see details in Table 2. We further collect a dataset of the same users from 30/12/2017 to 26/01/2018, ACST, as the ground truth for evaluation (*Darwin-eva*). It contains 47,749 statuses and 40,550 likes of 1,130 users. We then collect the *following* relations among those users. There are 58,503 relations among 3,880 users. From the interactions and following networks, we get the influence network. It includes 3,883 users and 58,799 edges (see Table 3).

In the *Darwin-eva* dataset, there are 13,656 tweets initiated by 858 users; 660 users who initiate 9,684 actions influence no one in the *Darwin* community; 198 users have 833 tweets out of their total 3,972 tweets that successfully influenced others. For these 833 tweets, on average, there are 2.80 users and 3.18 actions in an influence cascade, and the largest cascade contains 16 users and 20 actions. The average cascade size of users is smaller than the size of actions because each user can participate multiple times in a cascade. We focus on whom would be influenced, and the cascade size reported in the results is the number of users. More descriptions of the evaluation data will be provided in the results part.

Event-based Dataset-*"MelCup17"* We use Twitter *Streaming API* to track the terms "Melbourne Cup", "#MelCup" and "Derby-Day", from 00:00:00, 06/11/2017 to 23:59:59, 08/11/2017, Australia Eastern Standard Time (AEST), covering three days around the 2017 Melbourne Cup Day. 78,300 statuses messages posted by 34,573 users during the three days are collected. The results of Melbourne Cup came out on 15:05:19, 07/11/2017, AEST. The data before the results is used in the learning (*MelCup17-lrn*), and the data produced after the results is used for the evaluation (*MelCup17-eva*). We filter the users who have less than 6 statuses. As a result, 23,544 statuses posted by 1,484 users are investigated. The "likes" and *following* relations of the 1,484 users during this period are then collected. Descriptions of the *"MelCup17"* dataset are given in Table 2 and 3.

In the *MelCup17-eva* dataset, there are 3,194 tweets initiated by 740 users. 388 users who initiate 1,713 tweets influence no one in the *MelCup17* community. And 352 users have 1,052 tweets, out of their total 1,481 tweets, that successfully influence others. For the 1,052 tweets influencing others, the average cascade size of users is 5.5 and the largest cascade contains 89 users.

[1]The datasets and codes are available at https://bit.ly/2UlQ3xW

Table 4: Predicting size

Dataset	parameters	TC-Cp	TD-Cp	TDD-Cp
Darwin	α=1.2,t_s=15	1.1594	4.7386	0.7550
	α=1,t_s=15	**0.2227**	0.4954	**0.1665**
	α=0.8,t_s=15	0.2368	**0.2456**	0.1758
MelCup17	α=1,t_s=10	1.1869	9.1994	**0.4876**
	α=0.9,t_s=10	**0.6593**	1.6468	0.5030
	α=0.7,t_s=10	0.7012	**0.7186**	0.6343

6.2 Experimental Setup

In our experiments, T_0 is set as 300 minutes based on the observation that 80% of the reactions to the root action happen in 300 minutes.β is set as 0.5 and we choose p_1 as 0.001. The depth decay factor δ is learned as 0.15 and 0.13 for the "*Darwin*" and "*MelCup17*" respectively. Different t_s ranging from 3mins to 15mins are used. For different diffusion models, different scales of the learned $p^{T_0}(u, v)$s are used to adjust the sizes of the simulated influence cascades to fit the actual cascade sizes. The values of the scaling factor α are learned for different models. Under a set of parameters, we simulate the influence diffusion starting from each user who has at least an A_{u_TWT} in the evaluation data, with each model 100 rounds.

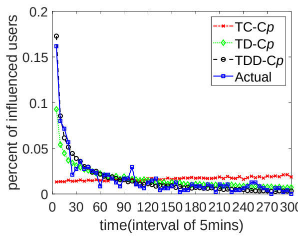
6.3 Experimental Results

Size Our analysis starts from the simulated cascade size $\hat{s}(u)$. The number of users a user influences when initiating an action is the most straightforward measure of how well the predictions fit what actually happened. We first figure out a proper α to make $\hat{s}(u)$ close to $s(u)$ in different models. Table 4 shows the NMSE values between $\hat{s}(u)$ and $s(u)$ under different α in TC-Cp, TD-Cp, and TDD-Cp models. With a same α, $\hat{s}(u)$ in the TD-C model is always the largest (see Section 4.2.2). For the *Darwin* dataset, when $\alpha = 1.2$, the $\hat{s}(u)$ values are generally larger than the $s(u)$, and the NMSEs are large. With $\alpha = 1$, the TC-Cp and TDD-C$_p$ predict the number of users a user will influence with low errors. And the TD-Cp model has similar performance when $\alpha = 0.8$. For the *MelCup17* dataset, the α is 0.9, 1 and 0.7 in the TC-C, TD-C, and TDD-C models respectively.

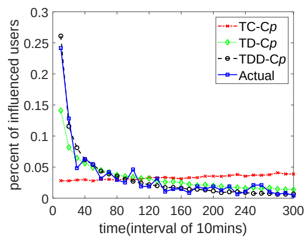
Shape Table 5 shows the statistics of the depth distribution of the influenced users with the three models ($t_s = 5mins$ for *Darwin*, $t_s = 3mins$ for *MelCup*17). The results for different t_s are similar. The prediction results with TC-C or TD-C models have a larger fraction of users influenced more than 1-depth away from the root user. The results with TDD-C model fit the depth distribution of influence cascades in the real world better than other models.

Table 5: Predicting shape

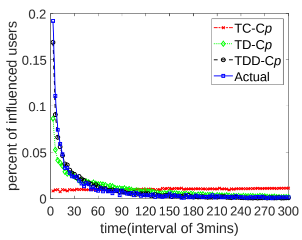
Dataset	Depth	Actual	TC-Cp	TD-Cp	TDD-Cp
Darwin	1	95.9%	86.82%	73.16%	94.64%
	2	2.4%	13.10%	26.05%	5.35%
	3	1.4%	0.08%	0.79%	0.01%
MelCup17	1	98.2%	68.67%	64.66%	95.13%
	2	1.8%	30.36%	33.96%	4.87%
	3	0.0%	0.97%	1.38%	0.00%



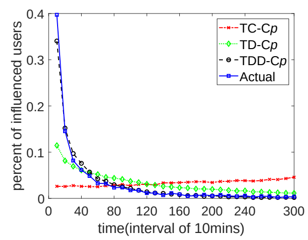
(a) *Darwin*, t_s=5mins



(b) *Darwin*, t_s=10mins



(c) *MelCup17*, t_s=3mins



(d) *MelCup17*, t_s=10mins

Figure 4: The time when users are influenced

Whom would be influenced by a particular person Table 6 shows the precision, recall and F-score values achieved with different models. For each A_{u_TWT}, its actual influence cascade is compared with 100 simulated cascades with each model, and the result with the best F-score is adopted. The values are the average of the results of all the A_{u_TWT}. Results with t_s of 5, 10, and 15 minutes in the *Darwin* dataset are given. The best performances achieved by the models are mainly with t_s=5mins, and the variations with different t_s are small. For brevity, we only provide the values with t_s=3min for the *MelCup17* dataset.

In general, the proposed models with p learned from both explicit and implicit interactions perform better than the models with p_e learned only from the explicit interactions. We provide a case study in *Darwin*: user $u2273$ and his follower $u2114$. In the learning data, $u2114$ has no explicit reactions to $u2273$, but 22 of $u2273$'s actions influence $u2114$ implicitly. In the evaluation data, $u2114$ is observed liking or retweeting eight of $u2273$'s messages. Without considering implicit influence, the models with very small $p_e(u2273, u2114) = 0.001$ can not predict that $u2114$ would be influenced by $u2273$ in most cases. With $p(u2273, u2114) = 0.152$, the models predict that $u2114$ is influenced by $u2273$ 20 times averagely in the 100 rounds of simulations. Our algorithm of learning implicit influence in Twitter based on *follow* relations, time and content similarity is effective.

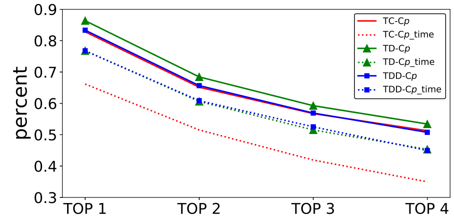


Figure 5: Top N predicted being influenced users in actual data

All the three types of models work in predicting whom would be influenced, referring to the left rows of the models in Table 6. In the *Darwin* data, when $t_s = 5mins$, the TDD-Cp model achieves the best precision of 0.7016. The highest recall goes to the TD-Cp model, which is 0.6584. The TDD-Cp model gives the best F-score of 0.6299. With different t_s, the TDD-Cp model always gives the best performance in precision and F-score measures, which shows that our proposed time and depth-sensitive diffusion model works. All our five models outperforms the baseline TC-Cp_e model.

When people would be influenced Figure 4 shows at what time the users are influenced in the evaluation data and the proposed models with different t_s. It shows the percentage of influenced users against the time that elapsed between t_v (the time when v is influenced) and t_u (the time when u initiate an action). The curves of each model's two variations are quite similar, and we only show the ones of models with p. In the actual situation, the number of influenced users decays over time. Both the TD-C and TDD-C models catch the decay behaviour. In terms of the percentage, the results of TDD-C fit the actual data better. The flat curve of TC-C model demonstrates that its prediction is not accurate.

The values of the prediction metrics calculated by considering when people are influenced are given in the right columns of different models in Table 6. To keep consistent in different t_s, the time related metrics are all calculated in a granularity of 30 minutes. We can see that the values of the prediction metrics of TC-C models drop significantly from above 0.6 to around 0.15. While the values with TD-C and TDD-C models also drop, to 0.40-0.50, they still show a good performance. In the *Darwin* dataset, when t_s=5mins, the precision, recall, F-score of the TDD-Cp model outperform the baseline TC-Cp_e model 219%, 244%, and 234% respectively. For the *MelCup17*, the numbers are 203%, 153%, and 186% when t_s=3mins. Our approach of modelling the influence diffusion in Twitter with time-sensitive probabilities is effective.

We are interested in the users who are influenced most times by a particular person. Figure 5 presents the percentage of the Top 1-4 predicted being influenced users that are actually influenced in the *Darwin* evaluation dataset with $t_s = 5min$. Looking at who are the most influenced user, the TD-Cp model performs slightly better than TC-Cp and TDD-Cp model. On average, 84.2% of the Top 1 predicted influenced users with the three models are actually influenced ; and the percentages are 66.4%, 57.6% and 51.8% for the Top 2, 3 and 4. Considering when the top users are influenced, the performance of TC-Cp model drops by 25%. While the performances of TD-Cp and TDD-Cp model drop by around 10%.

Table 6: Performance of predicting who will influence whom at what time

Data& metrics		TC-Cp		TC-Cpe (Baseline)		TD-Cp		TD-Cpe		TDD-Cp		TDD-Cpe	
		who	when	who	when	who	when	who	when	who	when	who	when
Darwin 5min	Precision	0.6618	0.1810	0.6581	0.1517	0.6593	0.4168	0.6454	0.4077	**0.7016**	***0.4845***	0.6910	0.4791
	Recall	0.6178	0.1627	0.6017	0.1251	0.6405	0.4023	**0.6584**	0.3912	0.6300	***0.4300***	0.6074	0.4122
	F-score	0.6079	0.1621	0.5978	0.1290	0.6107	0.3843	0.6082	0.3746	**0.6299**	***0.4303***	0.6159	0.4193
Darwin 10min	Precision	0.6759	0.1720	0.6635	0.1935	0.6637	0.4294	0.6368	0.4179	**0.7033**	***0.4750***	0.6888	0.4577
	Recall	0.6279	0.1506	0.6223	0.1628	**0.6647**	***0.4187***	0.6421	0.4008	0.6125	0.4147	0.6147	0.3978
	F-score	0.6171	0.1509	0.6094	0.1672	0.6208	0.3971	0.5990	0.3832	**0.6231**	***0.4185***	0.6187	0.4031
Darwin 15min	Precision	0.6809	0.1927	0.6579	0.2137	0.6569	0.4115	0.6522	0.4189	**0.6999**	***0.5539***	0.6773	0.5318
	Recall	0.6424	0.1707	0.6151	0.1935	**0.6666**	0.4066	0.6432	0.3962	0.6161	***0.4766***	0.6005	0.4611
	F-score	**0.6273**	0.1714	0.6060	0.1937	0.6150	0.3828	0.6061	0.3818	**0.6249**	***0.4884***	0.6058	0.4688
MelCup17 3min	Precision	0.5707	0.1539	0.5685	0.1445	0.5660	0.3840	0.5320	0.3365	**0.6112**	***0.4374***	0.6055	0.3891
	Recall	0.5485	0.1459	0.5310	0.1526	0.5520	0.3765	0.5393	0.3595	**0.5641**	***0.3861***	0.5133	0.3152
	F-score	0.4918	0.1282	0.4906	0.1263	0.4921	0.3308	0.4690	0.2976	**0.5223**	***0.3615***	0.4940	0.3032

7 CONCLUSION

In this paper we addressed the questions of measuring people's influence on others and predicting influence diffusion in Twitter. We proposed an effective algorithm for learning influence in Twitter. By capturing both the explicit and implicit influence in the action history, the pair-wise influence probabilities have been learned. We then proposed a time and depth-sensitive cascade model for influence diffusion by considering the temporal and structural features of actual influence cascades in Twitter. Experiments are conducted on two real-world Twitter datasets. Results show that our proposed models can predict the influence cascade in Twitter quite well, outperforming the existing models to predict when the users are influenced in terms of the balanced precision and recall. Our prediction task is important for understanding how influence diffusion works with many applications, such as marketing of time-sensitive goods and recommending the important messages that users are most likely to adopt in the information overloaded social networks.

Recent work has investigated topic-wise social influence diffusion [3]. It would be interesting to look at user-to-user influence and the spreading of influence in specific topics or domains.

REFERENCES

[1] Jeffery Ansah, Wei Kang, Lin Liu, Jixue Liu, and Jiuyong Li. 2018. Information Propagation Trees for Protest Event Prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 777–789.

[2] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. 2011. Everyone's an Influencer: Quantifying Influence on Twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*. ACM, New York, NY, USA, 65–74.

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2013. Topic-aware social influence propagation models. *Knowledge and information systems* 37, 3 (2013), 555–584.

[4] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, P Krishna Gummadi, et al. 2010. Measuring user influence in twitter: The million follower fallacy. *Icwsm* 10, 10-17 (2010), 30.

[5] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Synthesis Lectures on Data Management* 5, 4 (2013), 1–177.

[6] Manlio De Domenico, Antonio Lima, Paul Mougel, and Mirco Musolesi. 2013. The anatomy of a scientific rumor. *Scientific reports* 3 (2013), 2980.

[7] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. 2015. The structural virality of online diffusion. *Management Science* 62, 1 (2015), 180–196.

[8] Sharad Goel, Duncan J. Watts, and Daniel G. Goldstein. 2012. The Structure of Online Diffusion Networks. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12)*. ACM, New York, NY, USA, 623–638.

[9] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 241–250.

[10] Siwar Jendoubi, Arnaud Martin, Ludovic Liétard, Hend Ben Hadji, and Boutheina Ben Yaghlane. 2017. Two evidential data based models for influence maximization in twitter. *Knowledge-Based Systems* 121 (2017), 58–70.

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

[12] Konstantin Kutzkov, Albert Bifet, Francesco Bonchi, and Aristides Gionis. 2013. Strip: stream learning of influence probabilities. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 275–283.

[13] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. 2007. Patterns of cascading behavior in large blog graphs. In *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 551–556.

[14] Bjoern-Elmar Macek and Martin Atzmueller. 2013. Visualizing the impact of time series data for predicting user interactions. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 1477–1478.

[15] Yan Mei, Weiliang Zhao, and Jian Yang. 2017. Influence maximization on twitter: A mechanism for effective marketing campaign. In *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 1–6.

[16] Sancheng Peng, Yongmei Zhou, Lihong Cao, Shui Yu, Jianwei Niu, and Weijia Jia. 2018. Influence analysis in social networks: A survey. *Journal of Network and Computer Applications* (2018).

[17] Manuel Gomez Rodriguez, Jure Leskovec, David Balduzzi, and Bernhard Schölkopf. 2014. Uncovering the structure and temporal dynamics of information propagation. *Network Science* 2, 1 (2014), 26–65.

[18] Daniel M Romero, Brendan Meeder, and Jon Kleinberg. 2011. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, 695–704.

[19] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 67–75.

[20] Io Taxidou and Peter M. Fischer. 2014. Online Analysis of Information Diffusion in Twitter. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)*. ACM, New York, NY, USA, 1313–1318.

[21] Yu-Ting Wen, Wen-Chih Peng, and Hong-Han Shuai. 2018. Maximizing Social Influence on Target Users. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 701–712.

[22] 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 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1513–1522.