STRM: A Sister Tweet Reinforcement Process for Modeling Hashtag Popularity

Bidisha Samanta Abir De Niloy Ganguly
IIT Kharagpur, India
bidisha@iitkgp.ac.in
{abir.de,niloy}@cse.iitkgp.ernet.in

***Abstract*—With social media platform such as Twitter becoming the de facto destination for users' views and opinions, it is of great importance to forecast an information outbreak. In Twitter, tweets are often annotated with hashtags to help its users to quickly extract their contents. The existing approaches for modeling the dynamics of tweet-messages are usually limited to individual or simple aggregates of tweets rather than the underlying hashtags. In this paper, we develop, STRM, a novel point process driven model that considers the effect of cross-tweet impact in hashtag popularity. STRM, by assuming hashtag to be a heterogeneous collection of tweet-chains. Through extensive experimentation, we find that our algorithm - STRM, shows consistent performance boosts with six diverse real datasets against several strong baselines. Moreover surprisingly, it also offers significant accuracy gains in popularity-prediction for individual tweets as compared with the existing paradigms.**

I. Introduction

In Twitter, tweets are usually annotated with a hashtag for helping people to quickly retrieve their diverse contents. Thus accurate prediction of hashtag flow would help Twitter to rank content better, find trending hashtags faster. Altogether, it will enhance the Twitter service facilities by an improved content delivery system [1]. As a result, a considerable amount of research [1–12] on viral marketing and information cascade emphasizes the analysis of hashtag-popularity and its role in tweet propagation.

Research on hashtag-popularity predominantly follows two models. (i). Temporal or dynamical model. (ii) Static feature-based supervised model. The supervised approaches ([9, 10] and the citations therein) to compute hashtag-popularity heavily depend on feature-engineering. However, since hashtag-virality is inherently a temporal phenomenon, these models do not fit into online learning because the feature space they use is static in nature. Apart from this, these approaches do not take into account the self-exciting[1] nature of the hashtag diffusion. The class of temporal models [1–8] which are close to the work presented in this paper, primarily focuses on the propagation of individual tweets rather than the hashtags.

In this paper, we propose a technique to model hashtag popularity where a hashtag is associated with a collection of tweet-chains[2]. The popularity of a hashtag depends on two factors. (i). The intrinsic popularity/attractiveness of the hashtag. (ii). The popularity of its associated tweet-chains. However, these two factors are often convoluted with each other and hence, it is very difficult to decouple them with the temporal data. A hashtag often influences the flow of a tweet via the virality of her sister-tweets[3]. For example, a potentially insignificant tweet may suddenly go viral due to the influence of a popular sister-tweet. Hence, simply extending the prediction framework [1–8] designed for tweet to hashtag would not produce accurate result (experimental results also emphatically establish that).

In this context, we develop STRM: a **S**ister-**T**weet **R**einforcement **M**odel, to accurately capture the role of cross-tweet dependencies on the hashtag-propagation process. Our proposal, which is driven by a self exciting point process model, describes the collective tweet diffusion process by combining the self-exciting phenomenon with the sister-tweet influences in a principled way. Such an approach does not only capture the *heterogeneous dependencies* between tweets, but also *models a diverse type of tweets in a unified way*. For example, when a popular tweet is retweeted, it may hasten its own diffusion process, or speed up the flow of many rare-tweets or trigger new tweets. Indeed, our model can accurately capture all these three types of possible scenarios following a popular retweet-post. The proposed model implicitly considers the role of underlying network structure. But, unlike the existing works, we do not learn individual edge weights. Instead, the role of a graph is captured more macroscopically by the model. As a result, our model is *parsimonious, as it requires a few parameters to be fitted.*

In another departure from some of the existing works (e.g. [1]) that can only predict asymptotic popularity, our model also leads to a useful predictive formula to compute hashtag-popularity at *any point of time* in future. On six diverse datasets crawled from Twitter, our proposal offers *substantial accuracy gains beyond strong baselines* (Table II, Section V). In addition, we observe that our model performs well in wide range of sample size variation. More importantly, we found, though our model is designed to compute hashtag-popularity, it can also accurately predict the popularity of the individual tweets better than algorithms specifically catering to prediction of tweet popularity.

Contributions: Summarizing, the main contributions of our work are:

1) **An inter-tweet influence based model:** We develop a novel stochastic model for hashtag propagation that

[1]A tweet-stream is self-exciting if its own retweet-post reinforces the retweet process.

[2]A tweet-chain is a set consisting of a tweet and all its retweet instances.

[3]Two tweets are said to be the sisters of each other if they share a common hashtag.

captures the underlying cross-tweet reinforcement process (Section III). In this way, it captures both self-tweet and hashtag- tweet reinforcement processes.

2) **Parsimony:** Our model requires a few parameters to be fitted. It makes the model highly scalable.
3) **Capturing reality:** Despite the few parameters, the model still captures the complexity of the process. As a result, it mimics the reality by achieving significant accuracy boosts over the state-of-the-art methods.
4) **Forecasting ability:** Our model leads to a useful predictive formula that can forecast the hashtag popularity ideally in any point of time in future, even without the knowledge of the possible events going to come in-between.

II. Related works

Hashtag propagation and popularity analysis have been widely studied in different guises and formalized in recent years. Research on hashtag popularity traditionally follows two type of models. (i) Temporal models. (ii). Feature based supervised discriminative models.

A. Temporal Models

1) ***Reinforced Poisson Process (RPP)*** *[5, 6]:* Shen et al. in their paper [4], first exploited the concept of RPP in modeling popularity dynamics. Their model parameterizes the rate of posts $r(t)$ with three factors;- the intrinsic attractiveness of the post, the aging effect and the current number of posts. Later on, Gao et al. brought out a more sophisticated version [5] by taking a power-law relaxation function and the user-activity dynamics into account. A recent paper [6] considers the effect of network structure in RPP via a mixture process. Though such models are explainable as they unify three factors in a principled way, they are non-convex often resulting in an inefficient parameter-estimation. Moreover, in practice, they ignore the heterogeneous effects between tweets bearing same hashtag.

2) ***Hawkes Process and variants*** *[1, 2, 8, 13–15]:* The problem of non-convexity for RPP can be easily tackled with Hawkes process, a self exciting point process, that captures the bursty nature of information flow in the social media. Such models offer a change in the rate of posts with each new post, coupled with an aging factor. Recently, Zhao et al. developed a framework called "SEISMIC" [1], a variant of Hawkes process, in order to model the tweet-popularity. However, the predictive power of this model is only limited to computing asymptotic popularity, i.e. it can't predict the popularity at any given time t. Kobayashi et al. advocates incorporating circadian cycle in the dynamics of Hawkes process [2], since in some cases, tweet-posts actually follow such patterns.
Some of these models [13–15] advocate the role of network-topology in tweet-diffusion. However, for the hashtag propagation, the underlying networks are often disconnected, time-varying and huge, thus these models are difficult to learn.

3) ***Temporal pattern based models*** *[7, 16]:* These models originate following some observed patterns from temporal data For example, "RSC: Rest-sleep-comment model" [7] characterizes four type of patterns in the distribution of inter-arrival-times of postings. This model, however, fares poorly when a user frequently changes her behavior. In a similar spirit, the dynamical model in [16] aims to capture the rise and decay pattern of an event popularity from temporal data.

B. Feature based supervised models

The supervised approaches ([9, 10] and the citations therein) attach the popularity to temporal features and apply various supervised ML techniques (e.g. regression, classification etc.) to predict the future behavior. They primarily focus on finding the effective features, but the efforts needed to access these features are often huge. Moreover, these models essentially neglect the bursty behavior of temporal dynamics. Therefore, these models are not much effective to model hashtag dynamics.

Our work heavily relies on the temporal dynamics of hashtag propagation. To compare the utility and efficiency of our proposal, we use the ones in [1, 4, 8, 16] as the baseline-competitors (Section V-A).

III. Proposed Model

In this section, we first formulate our model of hashtag propagation, beginning with the data it is designed for, and then present methods for popularity prediction and model parameter estimation.

Temporal data: We define a tweet-chain as the set consisting of a tweet and all its retweet timestamps. Formally, a tweet-chain C can be written as, $C := T \times \{t_j |\ T \text{ is retweeted at time } t_j\}$. A hashtag $\boldsymbol{H}$ is formally defined as $\boldsymbol{H} := \{C_1, C_2, ..., C_N\}$, where C_i's are the tweet-chains. We record each timestamp of the (re)tweet-messages as $e_i = \{T[t_i], t_i\}$, meaning that at time t_i, a tweet $T[t_i]$ is (re)posted. Given a collection of retweets of all tweets of a hashtag $\boldsymbol{H}$, the history $\mathcal{H}(t) = \{e_1 = (T[t_1], t_1), \ldots, e_n = (T[t_n], t_n)\}$ gathers all messages of $\boldsymbol{H}$ upto but not including time t, i.e.,

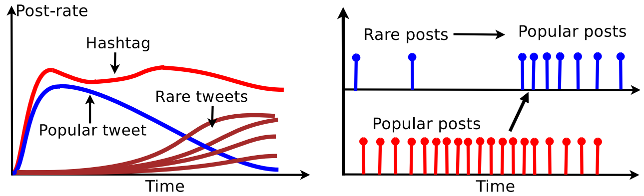
$$\mathcal{H}(t) = \{e_i = (T[t_i], t_i) \text{and}\, t_i < t\}, \tag{1}$$

Generative process for tweet-chains: We assume the (re)tweet-messages are being generated through a point-process model. When a tweet is posted, it evokes a retweet of the same and other tweets with the given hashtag. Naturally, the message times are represented by a counting process. In particular, given a hashtag $\boldsymbol{H}$, we denote the counting variable as $N(t)$, where $N(t) \in \{0\} \cup \mathbb{Z}^+$ counts the number of messages posted until and excluding time t_i. Then, we can characterize the message rate of the users using their corresponding conditional intensities or post-rates as

$$\mathbb{E}[dN(t) \,|\, \mathcal{H}(t)] = \lambda(t)\, dt, \tag{2}$$

where $dN(t)$ denotes the number of messages in the window $[t, t+dt)$ and $\lambda(t)$ denotes the associated user intensities, which may depend on the history $\mathcal{H}(t)$. Next, we specify the intensity functions $\lambda(t)$.

Intensity for messages: There is a wide variety of message intensity functions one can choose from to model the users' intensity $\boldsymbol{\lambda}^*(t)$ [17]. In this work, we emphasize two of the most popular functional forms used in the growing literature on social activity modeling using point processes [18, 19]:



a). Hashtag as a collection of heterogeneous tweets

b). Effect of cross-tweet reinforcement: rare tweets become popular later on.

Fig. 1: Role of tweets on hashtag propagation. Panel a) indicates most probable situations for a hashtag, where few tweets are very popular and most tweets are rare. Panel b) describes a cross-talk between tweets where a rare tweet becomes popular due to the presence of a popular sister tweet. This also shows the heterogeneous reinforcement process where a popular tweet-chain changes a rare-tweet dynamics but not the vice-versa.

I. **Poisson process.** The intensity is assumed to be independent of the history $\mathcal{H}(t)$ and constant, *i.e.*, $\lambda(t) = \lambda_0$.
II. **Hawkes processes.** The intensity captures a mutual excitation phenomenon between message events and depends on the whole history of message events $\mathcal{H}(t)$ before t:

$$\lambda(t) = \lambda_0 + \beta \sum_{t_i \in \mathcal{H}(t)} e^{-\omega_0 (t - t_i)} = \lambda_0 + \beta(\kappa(t) \star dN(t)) \quad (3)$$

where the first term, $\lambda_0 \geqslant 0$, models the initial publication-rate of messages, and the second term, with $\beta \geqslant 0$, assigns weight to the influence due to the publication of the previous messages. Here, $\kappa(t) = e^{-\omega_0 t}$ is an exponential triggering kernel modeling the decay of influence of the past events over time and $\star$ denotes the convolution operation.

A. STRM: A sister-tweet reinforcement model.

Our model mainly takes into account two crucial factors in modeling the dynamics of a hashtag. (i). The inherent popularity-dynamics which drives the spreading process of each individual tweet-chain. (ii). The mutual reinforcement process between sister-tweets bearing a same hashtag. However, the cross-tweet reinforcement is not homogeneous across tweets. For example, when a popular tweet is often re-tweeted, the dynamics of its sister-tweets gets more and more bursty, whereas, a rare-tweet hardly affects their dynamics. Such scenarios can be often captured by formulating a suitable triggering or intensity kernel $\kappa(t)$ in the Hawkes process (Eq. 3). In fact, $\kappa(t)$ should be non-uniform across the popularity distribution of tweets in a hashtag. That is, the kernel should be further parameterized by tweet-popularity k, to have $\kappa_k(t)$. In particular, $\kappa_k(t)$ should be chosen in such way that,

- For popular hashtags, i.e. when k goes high, $\kappa_k(t)$ pushes $\lambda(t)$ more towards a Hawkes process.
- For non-popular hashtags, $\kappa_k(t)$ goes low, suggesting a relatively small $\lambda(t)$.

We take

$$\kappa_k(t) = e^{-(\omega_0 + \frac{\omega}{k})t}$$

Hence the arrival rate of tweets can be written as:

$$\begin{aligned}\lambda(t; \boldsymbol{k}(t)) &= \lambda_0 e^{-\epsilon t} + \beta \sum_{t_i \in \mathcal{H}(t)} e^{-(\omega_0 + \frac{\omega}{k(t_i)})(t - t_i)} \\ &= \lambda_0 e^{-\epsilon t} + \beta \int_0^t e^{-(\omega_0 + \frac{\omega}{k(\theta)})(t-\theta)} dN(\theta) \quad (4)\end{aligned}$$

Here, $k(t_i)$ is the popularity of tweet posted at time t_i and $\boldsymbol{k}(t) := \{k(t_i) | t_i \in \mathcal{H}(t)\}$. Clearly, as k increases, the value of $\kappa_k(t)$ rises, resulting a large influence of a popular-tweets in other subsequent retweets. On the other hand, a retweet of not-so-popular tweet scarcely influences other tweets. An additional decay factor $e^{-\epsilon t}$ is incorporated in order to diminish the effect of initial condition, which we found to work well in practice.
Popularity distribution. In practice, given a hashtag, the distribution of the popularity of the individual-tweets usually follows a power-law:

$$p(k) = ck^{-\alpha} \text{ with, } c = \alpha - 1 \quad (5)$$

where k is the popularity of a tweet-chain.

Hence, the expected arrival rate of the process having tweet-chains with *random popularity* can be formulated as:

$$\tilde{\lambda}(t) = \mathbb{E}_{\boldsymbol{k}}[\lambda(t; \boldsymbol{k}(t))] = \int_1^\infty c\lambda(t; \boldsymbol{k}(t)) k^{-\alpha} dk \quad (6)$$

where, c is a constant given by Eq. (5).
Model Insights. Apart from considering the cross-tweet interactions, the proposed model offers following interesting insights.

(i) STRM models any *hashtag as a distributed tweet collection* in our proposal. It does not consider individual tweet-chain but an aggregate of tweet-chain in a more *non uniform* way.
(ii) STRM captures a *heterogeneous cross-tweet influence* via a popularity-parameterized triggering kernel (Eq. (4) and subsequent explanations). That said, the effect of a tweet-chain on its sister-chains varies across the popularity distribution. More is its popularity, more is this effect (Figure 1(b)).
(iii) Unlike prior work, STRM properly models *both popular and rare tweet propagation in a unified way* (Figure 1(a)). Consequently, such a unified approach accurately models the crosstalk between diverse tweet-dynamics, e.g. it can properly capture the phenomena where the retweet of a popular tweet may hasten its own diffusion process, and/or speed-up the flow of many rare-tweets and/or trigger new tweets. This makes STRM also an efficient predictive model for the popularity of individual tweets.
(iv) Another aspect of STRM is that it captures the hashtag dynamics by incorporating two reinforcement factors. (i). The self-exciting process of the individual tweets. (ii). The hashtag-tweet reinforcement process, which subsequently influences the hashtag dissemination.

B. Popularity forecasting

Our goal here is to develop efficient methods that leverage our model to forecast a hashtag's popularity at a given time t.

TABLE I: Statistics of hashtags collected

Events-type	Datasets	#Hashtags	Example-Hashtags	#Tweets	Total #Chains	Avg. chain-length
Entertainment	*Oscars*	8	#leonardodicaprio,#bearstory	10735	31	365
Entertainment	*MTV-Awards*	5	#mtvawardsstar,#blaonedirection	6427	277	23
Disaster	*Nepal-earthquake*	14	#kathmanduquake,#nepalrelief	42228	1182	35
E-Commerce	*BBD*	8	#aslidealsonebay,#bigbillionsale	7455	733	10
Sports	*Copa*	3	#copaamericaentd, #aocopa	8378	355	23
Sports	*ICCTw20*	12	#ICCWT20,#T20WorldCup	6652	10	665

In a similar spirit to [1], we measure popularity at time t, by the total number of retweet-counts for a hashtag. In the context of our model, we aim to compute $N^*(t) = \mathbb{E}_{\mathcal{H}(t)}[N(t)]$, the expected value of total retweet counts of all tweets. Here the expectation indicates the average of all random number of tweets, over the history from 0 to t.

$$N^*(t) = \mathbb{E}_{\mathcal{H}(t)}[N(t)] \tag{7}$$

$$= \mathbb{E}_{\mathcal{H}(t),\boldsymbol{k}(t)}\Big[\int_0^t \lambda(t;\boldsymbol{k}(t))dt\Big] = \mathbb{E}_{\mathcal{H}(t)}\Big[\int_0^t \tilde{\lambda}(t)dt\Big] \tag{8}$$

Theorem 1: Assume $t_0 := 0$, then the expected popularity of a hashtag is given by

$$\mathbb{E}_{\mathcal{H}(t)}\Big[\int_0^t \tilde{\lambda}(\tau)d\tau\Big] = \frac{1}{\epsilon}\lambda_0(1-e^{-\epsilon t}) + \lambda_0\int_0^t G(t)\star e^{-\epsilon t}dt \tag{9}$$

where $G(t) = \Big[\sum_{k=0}^{\infty}\frac{(bt^{(2-\alpha)})^k}{\Gamma((2-\alpha)k)}e^{-\omega_0 t}\Big]$ and $b = \frac{\omega_0^{(1-\alpha)}\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}$. It is interesting to note that $G(t) = \frac{d}{dt}E_{(2-\alpha)}(-bt^{(2-\alpha)})$, where $E_{(2-\alpha)}(-bt^{(2-\alpha)})$ is the *Mittag-Leffler* function of order $(2-\alpha)$ [20].

Proof: The proof is given in Appendix A.
In the following, we give the predictive formulas for the asymptotic behavior of a hashtag-popularity. The nature of asymptotic behavior is different in two cases: $\epsilon > 0$ and $\epsilon = 0$. We describe them in two separate lemmas.

Lemma 2: The expected asymptotic retweet count for a hashtag, where $\epsilon > 0$ is given by,

$$\lim_{t\to\infty}\mathbb{E}(N(t)) = \int_0^{\infty}\mathbb{E}(\lambda(t))dt = \frac{\lambda_0}{\epsilon}.\frac{\omega_0}{\omega_0 - \frac{\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}} \tag{10}$$

Proof: The proof is given in Appendix B.

Lemma 3: The trajectory of the expected asymptotic retweet count for a hashtag, where $\epsilon = 0$ is given by,

$$\lim_{t\to T}\mathbb{E}(N(t)) \to \frac{\lambda_0\omega_0 T}{\omega_0 - \frac{\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}} \tag{11}$$

Proof: The proof is given in Appendix C.

C. Parameter Estimation:

Given a collection of messages $\mathcal{H}(T) = \{(T_i, t_i)\}$ recorded during a time period $[0, T)$, our goal is to find the optimal parameters λ_0 and β by solving a maximum likelihood estimation (MLE) problem. We assign the decay parameters through cross-validation. Given that the $(i-1)$-th tweet has arrived at time t_{i-1}, the likelihood that the next tweet will come at time t_i can be expressed as follows:

$$p(t_i|t_{i-1}) = \lambda(t_i)e^{-\int_{t_{i-1}}^{t_i}\lambda(t)dt}$$

Hence the likelihood of the observed message $\mathcal{H}(T)$ is,

$$L(\lambda_0,\epsilon,\beta,\omega_0,\omega) = \prod_{i\in\mathcal{H}(T)} p(t_i|t_{i-1})$$

To this end, it is easy to show that the log-likelihood of the messages is given by

$$\log[L(\lambda_0,\epsilon,\beta,\omega_0,\omega)] = \sum_{t_i\in\mathcal{H}(T)}\log(\lambda(t_i)) - \int_0^T \lambda(t)dt$$
$$= -\frac{\lambda_0}{\epsilon}[1-e^{-\epsilon T}] + \sum_{t_i\in\mathcal{H}(T)}\int_1^k \log[\lambda_k(t)]k^{-\alpha}dk$$
$$-\beta\sum_{t_i\in\mathcal{H}(T)}\int_1^{\infty}\frac{k}{k\omega_0+\omega}(1-e^{-\omega_0(T-t_i)-\frac{\omega}{k}(T-t_i))}dk \tag{12}$$

Note that the MLE problem is concave and thus can be solved efficiently.

IV. Datasets

We implemented our proposal on hashtags associated with a diverse set of real events. The events are selected in such a way that they provide significant number of messages. Moreover, we took care of the choice of events, so that:

1) These events occurred in different time period and the duration of those events varies,
2) The associated hashtags have a variation in terms of the number of tweet chains associated with them.

Therefore we focus on popular events from entertainment, sports, e-commerce and disaster. We used Twitter search API to collect all the tweets (corresponding to a 2-3 weeks period around the event date) of the following events/topics, also summarized in Table I.

- **Oscars:** 2016 Academy Award ceremony, from Feb 24 to Feb 29, 2016. It has 8 hashtags with 31 chains. Some important hashtags of this event are, #leonardodicaprio, #bearstory.
- **MTV-Awards:** 2016 MTV Award Star program, collected from April 3 to April 12, 2016. It contains 5 hashtags with 277 tweet-chains. Couple of popular hashtags are, #mtvawardsstar, #blaonedirection .

- **Nepal-Earthquake:** Earthquake in Nepal, from April 25 to May 1, 2015. It contains 14 hashtags with 1182 chains. In this case, some hashtags like #kathmanduquake, #nepalrelief are actually used to speed-up counter-disaster managements.
- **BBD:** Bigbillionday sale of Flipkart India, from October 6 to October 8, 2014. It has 8 hashtags with 733 chains. Couple of popular hashtags for this event are, #aslidealsonebay, #bigbillionsale.
- **Copa:** Copa America Football tournament, from June 3 to June 26, 2016. It has 3 hashtags with 355 chains. The hashtags for this event are, #copaamericaentd, #aocopa.
- **ICCWT20:** ICC world cup T20, India, from March 8 to April 3, 2016. It has 12 hashtags with 10 chains. Couple of important hashtags are, #ICCWT20, #T20WorldCup.

V. Experiments

We compare STRM against several strong baselines which are state-of-the-art algorithms in the existing literature (see below). Apart from this, we also analyze the stability of our proposal through a wide variation of training-set size. Furthermore, to have a microscopic analysis of our approach, we carried out various experiments that reveal important insights about our model.

A. Baselines

We choose various types of baselines like RPP [4], simple Hawkes process [8], SEISMIC [1], SpikeM [16]. We also experimented with RSC [7] and NetRate [13], but because of their severely poor performances, we only present a comparative analysis with the first four models. The baselines are chosen in such a way, so that the performance of our model can be compared with a diverse class of existing temporal models. For example, RPP, Hawkes and SEISMIC represent the self-exciting processes, whereas, SpikeM and RSC are rather temporal pattern based approaches. These baselines have been primarily proposed for single tweet popularity prediction. We have extended them for hashtags by aggregating the corresponding tweets. An alternate way would have been to predict each individual tweet stream using the models and then aggregate the prediction. However, the models perform well only with popular tweets chains, but would make gross error with large number of rare tweet-chain present in the hashtag collection. An account of these paradigms is given in Section II, we here, briefly state their predictive mechanism.

1) Reinforcement Poisson Process, RPP [4]: Once the parameters are learned, the popularity can predicted as,

$$c^d(t) = (m + n_d)e^{\lambda^*_d(F_d(t;\theta_d*) - F_d(T;\theta^*_d))} - m$$

where, m is the effective number of posts for all hashtags, n_d is the number of posts for a particular tweet received, $\lambda_d(t)$ is an intrinsic attractiveness of an item, $F_d(t;\theta_d) = \int_0^t f_d(t;\theta_d)dt$ with $f_d(t;\theta_d)$ being a general temporal relaxation function, and $i_d(t)$ is a reinforcement factor.

2) Hawkes Process [8]: To replicate this model, we use the aggregate of tweets as input to this model. Given a Hawkes process has the form,

$$\lambda(t) = \mu + \alpha \int_{-\infty}^{t} \exp(-\beta t) dN(t)$$

. The prediction function can be written as

$$N(t) = \frac{\lambda(t_n)}{\alpha-\beta}(e^{(\alpha-\beta)t} - 1) + \frac{\beta\mu}{\alpha-\beta}\Big[\frac{e^{(\alpha-\beta)t} - 1}{\alpha-\beta} - t\Big]$$

3) SEISMIC [1]: This model estimates the final size of an information cascade given the posting timestamps and the follower counts of users who are posting. It is a variant of Hawkes process characterizing posts by their infectiousness termed as a measure of re-share probability. The prediction function is defined as

$$\hat{R}_\infty(t) = R_t + \alpha_t \frac{\hat{p}_t(N_t - N^e_t)}{1 - \gamma_t \hat{p}_t n_*}, 0 < \alpha_t, \gamma_t < 1 \qquad (13)$$

where p_t is the infectiousness, N^e_t - effective cumulative degree of re-sharers by time t, N_t - cumulative degree of re-sharers by time t.

4) SpikeM [16]: This approach generalizes the empirical observations of temporal behavior of popularity in social media. SpikeM captures the behavior of the users as the following equations

$$\Delta B(n+1) = p(n+1).(U(n)\sum_{t=n_b}^{n}(\Delta B(t) + S(t)).f(n+1-t) + \hat{\epsilon})$$

$B(n)$ and $U(n)$ refer to the number of affected and unaffected users at time t_n respectively. At any given time instant, the model predicts the number of infected users indicating the number of users who have tweeted on an event. The number of attention received is estimated as the number of people affected in this process.

B. Metrics

We use the following performance metrics for our proposal and all the baselines.

1) Mean Absolute Percentage Error (MAPE): It is a commonly used measure of the mean deviation between the observed and the predicted popularity for a hahstag up to time t. It is defined by the formula

$$\text{MAPE}(\boldsymbol{H}) = \frac{1}{M_{\boldsymbol{H}}}\sum_{i=0}^{M_{\boldsymbol{H}}-1}\left|\frac{\hat{N}_{\boldsymbol{H}}(t_i) - N_{\boldsymbol{H}}(t_i)}{N_{\boldsymbol{H}}(t_i)}\right|.$$

Here $\hat{N}_{\boldsymbol{H}}(t)$ and $N_{\boldsymbol{H}}(t)$ are the estimated and actual number of retweets for a hashtag $\boldsymbol{H}$ respectively, at time t. For a given dataset, we report MAPE as the average of all MAPE($\boldsymbol{H}$) over the hashtags $\boldsymbol{H}$ of that dataset.

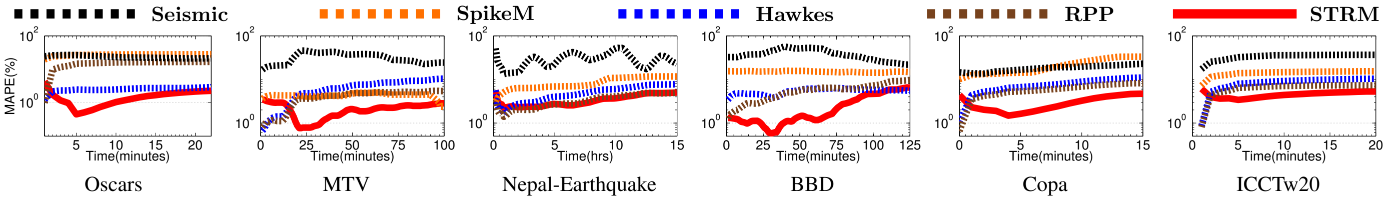


Fig. 2: Variation of popularity forecasting performance with time using a 20% held-out set for each real-world dataset. Time-scales are given minutes except Nepal-Earthquake dataset which is given in hours. Performance is measured in terms of Mean Absolute Percentage Error (MAPE) of the total retweet count.

2) Accuracy: This is a measure that the fraction of retweet count correctly predicted for a hashtag given error tolerance ϵ_1. Formally,

$$\text{Accuracy}(\boldsymbol{H}) = \frac{1}{M_{\boldsymbol{H}}} \sum_{i=0}^{M_{\boldsymbol{H}}-1} \mathbf{1}\Big[\left| \frac{\hat{N}_{\boldsymbol{H}}(t_i) - N_{\boldsymbol{H}}(t_i)}{N_{\boldsymbol{H}}(t_i)} \right| < \epsilon_1 \Big].$$

Here $\mathbf{1}(.)$ is the indicator function. Like MAPE, we report *accuracy* as the average of all *accuracy*$(\boldsymbol{H})$ over the hashtags $\boldsymbol{H}$ of that dataset.

TABLE II: MAPE (%) of proposed and baseline algorithms on all datasets with 80% training-set.

Datasets	STRM	RPP	Hawkes	SEISMIC	SpikeM
Oscars	**10.25**	18.30	*10.80*	17.34	27.22
MTV-Awards	**2.69**	11.17	*9.74*	31.32	22.40
Nepal-earthquake	**4.91**	11.58	*6.13*	13.89	35.27
BBD	**5.06**	16.72	*5.34*	29.12	5.16
Copa	**4.68**	8.58	*7.34*	24.11	20.34
ICCTw20	**5.05**	*8.61*	9.63	32.01	14.73

C. Performance Comparison - MAPE

As we mentioned earlier, we compared our method with four baselines. Table II dissects a comparative sketch in terms of MAPE with 80% training-set size. We observe that for all datasets, STRM gains substantial MAPE against all baselines. Also in terms of accuracy, our algorithm gives much better results for most of the datasets.

The performance of SEISMIC is consistently poor. SEISMIC assumes that the spreading rate of a tweet is proportional to the out-degree of the node which it is being exposed to. As a result, whenever a high-degree user views a tweet, SEISMIC assigns it a high probability of getting popularized. However, this may not be true in practice. For example, a inconsequential message may not be further retweeted even if it is introduced by a very influential node. Therefore, SEISMIC fails to properly model the dynamics of rare tweets. As a result, it cannot accurately predict the popularity of hashtags that usually contain a large number of rare tweets. The poor performance of SEISMIC can also be attributed to its limited predictive mechanism;- it can only provide an estimate of asymptotic popularity. That's why when tested on a small held-out temporal data, the model fares quite poorly. Apart from these, the nature of α_t (Eq. (13)) also makes the model weak in terms of predictive performance. The choice of α_t relies on the assumption that the posts get stale and outdated as time passes. This assumption immediately rules out the bursty nature of the data;-which is an important characteristics of any message-streams of social media.

The performance of SpikeM is better than SEISMIC for most of the datasets. This is due to the fact that SpikeM does not use user-specific information. Consequently, it is less over-fitting than SEISMIC. But this model does not perform well w.r.t. the other baselines e.g. Hawkes or RPP. This is because, this model emphasizes on modeling realistic patterns from the temporal data (e.g. periodicity). However, the temporal pattern in training-set may not (usually does not) match with that in the test-set. Such a non-invariance nature of temporal datasets often restricts the model from foreseeing the long-term nature in the hashtag-dynamics. For example, this model cannot capture a situation when a unpopular hashtag becomes popular later on. In fact, a more careful scrutiny reveals that spikeM is useful only for short-term forecasting [16, Section 7]. Apart from that, its predictive power is further limited since it can capture only the tail-part of the trajectory of posts [16, Section 4.4]. Indeed, the model emphasizes to capture a set of temporal patterns, forecasting ability is not its primary objective.

The performance of RPP is relatively better than SEISMIC and SpikeM. It does not function based on a fixed set of patterns from the temporal data. Unlike SEISMIC, it does not over-emphasize the role of individual user information. Instead, they attempt to capture a more fundamental property of the data, the self exciting nature which is often one of the major characteristics of the tweet-streams. In fact, it also takes into account the individual tweet-attractiveness, a key factor, however, often ignored by the other models. Moreover, it attempts to capture the aging of the posts too. Despite these strengths, RPP has several drawbacks that makes it a weaker model than STRM and Hawkes. First, it treats hashtag as homogeneous aggregates of tweets. As a result, it fails to capture the effect of the inherent popularity of the hahstag and the way it impacts the popularity of constituent tweets. Apart from this, the non-convexity of the learning problem is another crucial problem of RPP. This makes the actual parameters difficult to estimate and consequently the parameters are often unidentifiable.

The performance of Hawkes process is better than RPP; it is the second best proposal among all we have experimented with. There are mainly two distinctive features that help Hawkes to obtain a significant performance-boost w.r.t. RPP. First, Hawkes process takes into account the temporal effect of each and every post rather than their simple collective effect. This often helps Hawkes to capture better the bursty nature present in the datasets. Apart from that, underlying the learning problem is convex for this approach. As a result, one can accurately estimate the parameters, making it a robustly identifiable learning model. However, like all the existing

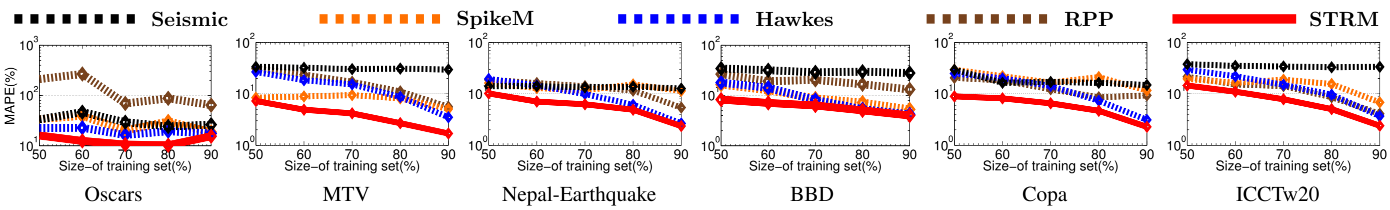


Fig. 3: Effect of training set size on the MAPE

models, it fails to capture the real hashtag dynamics as it ignores the cross-tweet dependence.
On the other side, STRM properly models the cross-talk between tweets to accurately record the effect of inherent attractiveness of the hashtags. It also captures the self-exciting phenomenon of the tweet-streams like Hawkes and RPP. Moreover, our model is convex and parsimonious. These factors, together, help our model to achieve significant improvement for almost all datasets.

D. *Forecasting Capability*

In order to have a better understanding of the performance of all the models, we further probed into regular time-stamps in the 20% held-out set and computed MAPE for each sample point. Figure 2 shows the change of predictive-performance of popularity with time (MAPE of sample points aggregated over time). We observe the forecasting performance of STRM is better than the other methods. However, as time elapses, the performance of all the approaches degrades. It is because, as time increases, the task of predicting popularity becomes more and more difficult, as one misses the possible important signals that appears in-between. However, the degradation-rate is relatively lower in RPP, Hawkes and STRM because unlike the other proposals, they can accurately compute the expected future-events. Moreover, the consideration of heterogeneous cross-tweet dependencies further boosts the performance of STRM.

E. *Stability to sampling*

Figure 3 shows the variation of MAPE with training-set size. We observe that almost all the algorithms show improvement in performance with increase in training-set size. It can be observed that the performance of STRM quickly improves as the training-set size increases, whereas, the rate of decrease of MAPE is relatively low for all competitors. In fact the performance variation of SEISMIC is almost steady throughout the variation of training-set size. Since SEISMIC can only compute the asymptotic popularity, it cannot accurately compute the popularity of any hashtag in any finite-time.

F. *Variation of performance with length of tweet-chains*

Figure 4 shows that the performance of STRM is best across hashtags whether we sort them with respect to the largest or average number of tweet-chain it contains. We observe, as the length of the largest chain increases (Figure 4(a)), the performance of all the baselines (except Hawkes) degrades. Perhaps surprisingly, RPP fares poorly for popular hashtags. A careful scrutiny reveals that, in case of RPP, the rate of post is directly proportional to the number of previous posts, but not the time of each post. As a result, as time goes, RPP has a tendency of over-estimating the count of popular-tweets. Although, RPP models the aging effect, for highly popular posts, the effect cannot offset the error which gradually enters into the system over time. However, it is overcome by STRM and Hawkes, since both of them assign an aging factor to each post. The cross-tweet reinforcement factor gives our proposal a further boost, which results in its better performance than Hawkes.

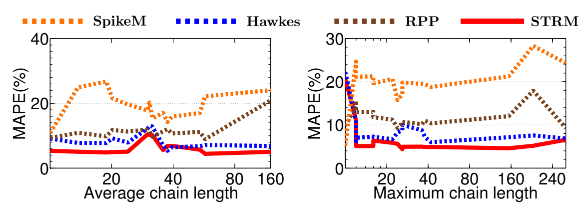


Fig. 4: Effect of length and size of tweet-chains on the MAPE

G. *Variation of performance with change of λ of training and testing*

In real-life temporal data, the distribution of tweet-posts are bursty and full of irregularities. The learning models are usually favored by two similar distributions in training and test set. However, as we said, such assumption is not valid in a real-life temporal dataset. As a result, a predictive model should be enough robust to capture the irregularities of the datasets. Figure 5A depicts the variation of performance with the difference (δ) in post-rates in training and test set. We observe that the performance of SpikeM is very poor. It is because, SpikeM attempts to learn from a set of fixed patterns of the training-set. However, this set of patterns may not be present in the test-set in most cases. Our model not only performs better across a wide variation of δ, but also it is consistently steady which indicates that STRM picks up important signals from the training data.

TABLE III: Accuracy (%) of proposed and baseline algorithms on all datasets with 80% training-set.

Datasets	STRM	RPP	Hawkes	Seismic	SpikeM
Oscars	**80**	63	*80*	59	78
MTV-Awards	**95**	*92*	65	40	84
Nepal-earthquake	**99**	80	*94*	55	70
BBD	*94*	53	**99**	60	83
Copa	**94**	74	*84*	40	49
ICCTw20	**88**	*84*	70	35	70

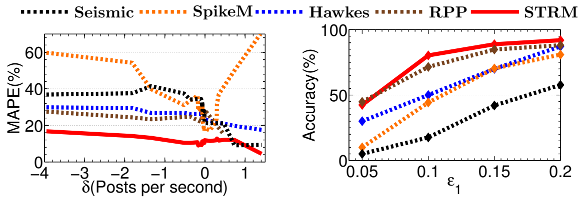


Figure (5A): MAPE variation with respect to the change in rate of arrival and testing

Figure (5B): Change of accuracy with increase of ϵ_1 for ICCTw20

TABLE IV: Performance of top-5 popular tweet-chains across all the datasets

		Learning parameters from hashtag		Learning parameters from individual chain	
Datasets	**STRM**	**Hawkes**	**RPP**	**Hawkes**	**RPP**
BBD	**0.158**	*0.491*	*1.212*	2.231	1.795
Oscars	**0.102**	0.630	*0.110*	*0.123*	0.282
Nepal-earthquake	0.369	*0.234*	***0.086***	1.207	0.099
MTV-Awards	**0.142**	0.293	5.194	0.105	4.414

H. Performance comparison in terms of Accuracy

We have also reported accuracy by taking the error tolerance (ϵ_1) as 0.15 across all the baselines in Table III. For five out of six datasets, STRM performs best among all competitors. We also observe that with respect to this metric (unlike MAPE) RPP gives Hawkes a very fair competition and as a result, in MTV-Awards and ICCTw20 datasets, it performs much better than Hawkes. This means for a reasonable number of hashtags, RPP is being able to predict their popularities within a tolerable limit. However, for cases when RPP fails to accurately predict, it gives very poor predictions and as a result, the performance of RPP is consistently poor w.r.t Hawkes if the performance is measured in terms of MAPE. We further vary the value of ϵ_1 and find that the trend of performance of various methods remains roughly similar across all dataset. For reference the result for dataset ICCTw20 is reported as Figure 5B.

I. Prediction performance in popular tweet-chains

In this subsection, we turn our attention from hashtag and try to predict the count of most popular tweets within a hashtag. As we know STRM does not explicitly compute the popularity of a tweet-chain but derives the spread (denoted by $N(t)$) of the underlying hashtag. From $N(t)$ one can approximately find the popularity of a tweet-chain C as $N_C(t) = p_C \cdot N(t)$. Here p_C is the fraction of tweets of the tweet-chain (C) present in the training-set. The parameters of both Hawkes and RPP are assigned following two different protocols: (i). assigning same set of parameters to all tweet-chains within a hashtag. and (ii). assigning different parameter-sets to each individual chain.

Table IV provides a comparative performance analysis of top-5 tweet-chains in terms of popularity across all datasets. Since, the existing approaches cannot work on rare tweets due to insufficient training points, this analysis is limited to only popular tweets. report the average MAPE over top 5 popular tweet-chains from all hashtags in the dataset. We find that although the primary objective of STRM is to compute the popularity of hashtag, it performs significantly well even in predicting the popularity of individual-tweet-chain. In this context, Table IV shows that the method, wherein parameters corresponding to each tweet-chain are learned, generally performs the worse. Its performance is particularly poor in case of BBD and Nepal-earthquake datasets. It is because, for both these datasets, the users mainly relied on Twitter for getting news updates. As a result, many tweet-posts in these two datasets triggered various diverse kind of reaction tweets. Thus, simply assigning many parameters to individual tweet-chains without considering the entire eco-system, leads to over-fitting of such models. STRM accurately models this inter-tweet interactions and therefore performs better in most of the cases. However, for Nepal earthquake, STRM fails to beat Hawkes for top 5 popular tweet-chains. It is because, this particular event was more topic-centric and we believe, a more generalized version of STRM needs to be derived that will consider cross-tweet dependencies from different hashtags of the same topic.

VI. Conclusion

In this paper, we develop, STRM, a novel point-process driven framework, which takes into account the cross-tweet reinforcement process for modeling hashtag dynamics. This model unifies the self-exciting nature of the individual tweets and crosstalk between sister-tweets in a principled way. Such a unified approach does not only accurately predict a hashtag popularity for which it is designed, but also gives a good measure of the popularity of individual tweets. Through rigorous experimentation, we validate the utility of our approach against six state-of-the-art algorithms. We also provide a detailed explanation behind this improvement. We observe that STRM performs well across a wide range of training-set size, indicating the stability of our proposal. As a future work, one may extend this work to model the building process of a story in various social media. It would help to understand the dynamics of trending topics in Twitter, the metamorphosis of articles in various forums, wikipedia etc.

Appendix

A. Proof of Theorem 1

Proof: From Eq. (6), we have

$$\tilde{\lambda}(t) = \lambda_0 e^{-\epsilon t} + c\beta \int_1^\infty \int_0^t e^{-(\omega_0 + \frac{\omega}{k})(t-\theta)} k^{-\alpha} dN(\theta) dk \tag{14}$$

Changing the order of the integration of the second part

$$\tilde{\lambda}(t) = \int_1^\infty c k^{-\alpha} \lambda_0 e^{-\epsilon t} dk + c\beta \int_0^t \int_1^\infty e^{-(\omega_0 + \frac{\omega}{k})(t-\theta)} k^{-\alpha} . dk dN(\theta) \tag{15}$$

Integration with respect to k of the second term is given by,

$$\int_1^\infty e^{-(\omega_0 + \frac{\omega}{k})(t-\theta)} k^{-\alpha} dk \approx e^{-\omega_0 (t-\theta)} \int_0^\infty e^{-\frac{\omega}{k}(t-\theta)} k^{-\alpha} dk$$
$$= e^{-\omega_0 (t-\theta)} (\omega(t-\theta))^{1-\alpha} \Gamma(\alpha - 1)$$

Hence Eq. (15) becomes,

$$\tilde{\lambda}(t) = \lambda_0 e^{-\epsilon t} + a\int_0^t e^{-\omega_0(t-\theta)}(\omega(t-\theta))^{1-\alpha} dN(\theta) \quad (16)$$

where $a = c\beta\Gamma(\alpha-1) = \beta\Gamma(\alpha)$.
Now since, $\mathbb{E}(dN(\theta)|\mathcal{H}(\theta), \boldsymbol{k}(\theta)) = \lambda(\theta)d\theta$, we have, from Eq. (16)

$$\begin{aligned}\mathbb{E}_{\mathcal{H}(t)}(\tilde{\lambda}(t)) &= \lambda_0 e^{-\epsilon t} \\ &+ a\int_0^t e^{-\omega_0(t-\theta)}(\omega(t-\theta))^{1-\alpha}\mathbb{E}_{\mathcal{H}(\theta)}(\tilde{\lambda}(\theta))d\theta \quad (17)\end{aligned}$$

Taking Laplace transform (LT) with Laplace variable s, we get

$$\Lambda(s) = \frac{\lambda_0}{s+\epsilon} + \frac{a\omega^{1-\alpha}\Gamma(2-\alpha)}{(s+\omega_0)^{2-\alpha}}\Lambda(s) \quad (18)$$

where $\Lambda(s)$ is the LT of $\mathbb{E}_{\mathcal{H}(t)}(\tilde{\lambda}(t))$. From eq. (18), we have,

$$\Lambda(s) = \frac{\lambda_0}{s+\epsilon}\frac{(s+\omega_0)^{2-\alpha}}{(s+\omega_0)^{2-\alpha}-b} \quad (19)$$

where $b = (a\omega^{1-\alpha}\Gamma(2-\alpha)) = \frac{\omega_0^{1-\alpha}\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}$ (using the relation between gamma and sine functions).

which gives, $\tilde{\lambda}(t) = \lambda_0 e^{-\epsilon t} + \lambda_0\Big[\sum_{k=0}^{\infty}\frac{(bt^{(2-\alpha)})^k}{\Gamma((2-\alpha)k)}e^{-\omega_0 t}\Big] \star e^{-\epsilon t}$

It directly concludes Eq. (9)

■

B. Proof of Lemma 2

Proof: Equation (18) directly gives $\lim_{t\to\infty}\mathbb{E}(N(t)) = \int_0^\infty \mathbb{E}(\lambda(t))dt$=$\lim_{s\to 0} s\left(\frac{\Lambda(s)}{s}\right) = \frac{\lambda_0}{\epsilon}.\frac{\omega_0}{\omega_0 - \frac{\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}}$ ■

C. Proof of Lemma 3

Proof: Equation (18) gives $\lim_{t\to\infty}\mathbb{E}(N(t)) = \int_0^\infty \mathbb{E}(\lambda(t))dt$=$\lim_{s\to 0} s\left(\frac{\Lambda(s)}{s}\right) = \lim_{s\to 0}\frac{\lambda_0}{s}.\frac{\omega_0}{\omega_0 - \frac{\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}}$ which, from Tauberian theorem [21], gives $\frac{\lambda_0\omega_0 T}{\omega_0 - \frac{\beta(\alpha-1)\pi}{\sin((\alpha-1)\pi)}}$

■

Acknowledgements: Abir De is partially supported by Google India under the Google India PhD Fellowship Award. The authors would also like to thank Information Technology Research Academy (ITRA) for supporting this work.

References

[1] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "Seismic: A self-exciting point process model for predicting tweet popularity," in *KDD*, 2015.

[2] R. Kobayashi and R. Lambiotte, "Tideh: Time-dependent hawkes process for predicting retweet dynamics," in *AAAI*, 2016.

[3] A. Kupavskii, L. Ostroumova, A. Umnov, S. Usachev, P. Serdyukov, G. Gusev, and A. Kustarev, "Prediction of retweet cascade size over time," in *CIKM*, 2012.

[4] S. Hua-Wei, D. Wang, C. Song, and A. Barabasi, "Modeling and predicting popularity dynamics via reinforced poisson processes," in *AAAI*, 2014.

[5] G. Shuai and C. Jun, Ma Zhumin, "Modeling and predicting retweeting dynamics on microblogging platforms," in *WSDM*, 2015.

[6] J. Gao, H. Shen, S. Liu, and X. Cheng, "Modeling and predicting retweeting dynamics via a mixture process," in *WWW*, 2016.

[7] A. F. Costa, Y. Yamaguchi, A. J. M. Traina, C. T. J. 1, and C. F. 3, "Rsc: Mining and modeling temporal activity in social media," in *KDD*, 2015.

[8] P. Bao, H.-W. Shen, X. Jin, and X.-Q. Cheng, "Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes," in *WWW*, 2015.

[9] N. Rosenfeld, M. Nitzan, and A. Globerson, "Discriminative learning of infection models," in *WSDM '16*.

[10] S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari, "Learning social network embeddings for predicting information diffusion," in *WSDM '14*.

[11] M. R. Rahman, J. Han, and C.-N. Chuah, "Unveiling the adoption and cascading process of osn-based gifting applications," in *IEEE INFOCOM*, 2015.

[12] D. Wang, H. Park, G. Xie, S. Moon, M.-A. Kaafar, and K. Salamatian, "A genealogy of information spreading on microblogs: A galton-watson-based explicative model," in *IEEE INFOCOM*, 2013.

[13] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the Temporal Dynamics of Diffusion Networks," in *ICML*, 2011.

[14] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez, "Learning and forecasting opinion dynamics in social networks," in *NIPS, 2016*.

[15] A. De, S. Bhattacharya, P. Bhattacharya, N. Ganguly, and S. Chakrabarti, "Learning a linear influence model from transient opinion dynamics," in *CIKM, 2014*.

[16] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *WWW*, 2015.

[17] O. Aalen, Ø. Borgan, and H. Gjessing, *Survival and event history analysis: a process point of view*. Springer Verlag, 2008.

[18] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, L. Song, and H. Zha, "Shaping social activity by incentivizing users," in *NIPS*, 2014.

[19] I. Valera and M. Gomez-Rodriguez, "Modeling adoption and usage of competing products," in *Proceedings of the 2015 IEEE International Conference on Data Mining*, 2015.

[20] I. Podlubny, *Fractional differential equations*. Academic press, 1998.

[21] W. Feller, *An introduction to probability theory and its applications: volume II*, 1968.