SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity

Qingyuan Zhao
Stanford University
qyzhao@stanford.edu

Murat A. Erdogdu
Stanford University
erdogdu@stanford.edu

Hera Y. He
Stanford University
yhe1@stanford.edu

Anand Rajaraman
Stanford University
anand@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

Social networking websites, like Twitter and Facebook, allow users to create and share content. As users of these sites reshare others' posts with their friends and followers, big information cascades of post resharing can form. One of the central challenges in understanding such cascading behaviors is in forecasting information outbreaks, where a single post becomes widely popular by being reshared by many users.

In this paper, we focus on predicting the final number of reshares of a given post. We build on the theory of self-exciting point processes to develop a statistical model that allows us to predict the final number of reshares of a given post that is being reshared through the network. Our model requires no training or expensive feature engineering and results in a simple and efficiently computable formula that allows us to in real-time answer questions like: Given the post's resharing history so far, what will be its final number of reshares? And, which will be the most reshared posts in the future? We validate our model on one month of complete Twitter data and demonstrate a strong improvement in predictive accuracy over existing approaches. Our model gives only 15% relative error in predicting final size of an information cascade after observing it for just one hour.

Categories and Subject Descriptors: H.2.8 **[Database Management]**: Database applications—*Data mining*
General Terms: Algorithms; Experimentation.
Keywords: tweet popularity; information diffusion; self-exciting point process; social media.

1. INTRODUCTION

Online social networking services, like Facebook, Youtube, and Twitter, allow their users to post and share content in the form of posts, images, and videos [9, 16, 20, 29]. As a user is exposed to posts of others she follows, the user may in turn reshare a post with her own followers, who may further reshare it with their respective sets of followers and this way large information cascades of post resharing behavior may spread through the network.

A fundamental question in modeling information cascades is to predict future evolution of a cascade. Arguably the most direct way to formulate this question is to consider predicting the final size of the information cascade. That is, to predict how many reshares a given post will ultimately receive.

Predicting the attention or the ultimate popularity of a content is important for content ranking and aggregation. For instance, Twitter is overflowing with posts and users have a hard time keeping up with all of them. Thus, much of the content gets missed and eventually lost. The ability to predict the ultimate popularity of content would allow Twitter to better rank content, faster discover trending posts, and improve the content placement in content-delivery networks. Moreover, predicting the size of information cascades allows us to gain fundamental insights into predictability of collective behaviors where uncoordinated actions of many individuals lead to big spontaneous outcomes, for example, large information outbreaks.

Most research on predicting information cascades involves extracting an exhaustive set of features describing the past evolution of a cascade and then using these features in a simple machine learning classifier to make a prediction about the future growth of the cascade [4, 6, 16, 19, 25, 29]. However, feature extraction can be expensive and cumbersome, and one is never sure if more effective features could be extracted. The question thus remains whether it is possible to design a simple and principled bottom-up model of cascading behavior. The challenge lies in defining a model of an individual's behavior and then aggregating the effects of the individuals in order to make a good global prediction.

Present work. Here we focus on predicting the final size of an information cascade spreading in a social network. We develop a statistical model based on the theory of *self-exciting point processes*. A point process is also called a *counting process* when it is indexed by time, which counts the number of instances (reshares, in our case) over time. In contrast to homogeneous Poisson processes which assume that the intensity of the process is constant over time, self-exciting processes assume that all the previous instances (*i.e.*, reshares) influence the future evolution of the process. Self-exciting point processes are frequently used to model "rich get richer" phenomenon [21, 22, 32, 35], and are ideal for modeling information cascades in networks, because every new reshare of a post not only adds 1 to its cumulative reshare count, but also exposes new followers, who may further reshare the post.

We develop SEISMIC (*Self-Exciting Model of Information Cascades*) for predicting the total number of reshares of a given post. In our model, each post is fully characterized by its *infectiousness*, which models post's resharing probability as a function of time.

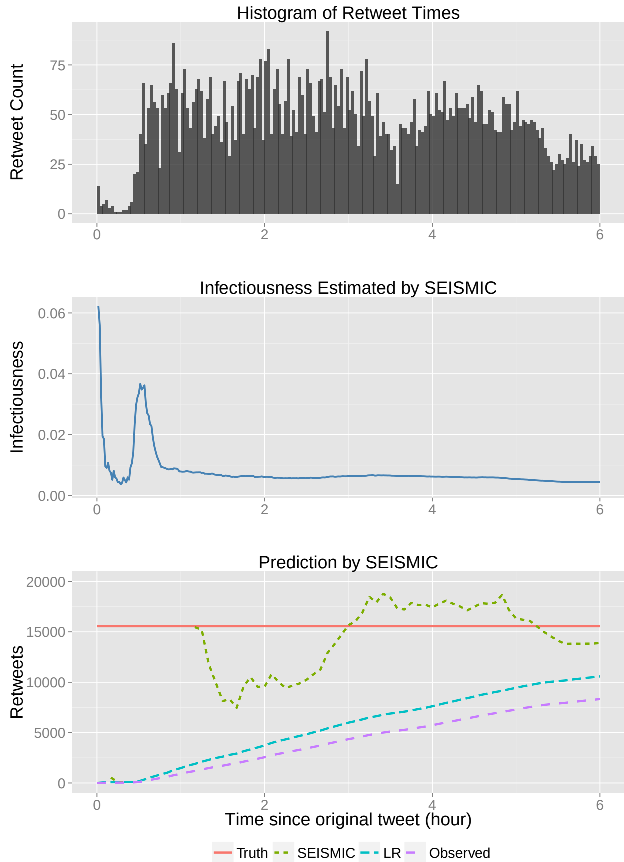

Figure 1: Retweeting activity in the first 6 hours of a popular tweet [1] (top). The tweet relates the death of dictator Muammar Gaddafi with singer Justin Bieber. Interestingly, the car manufacturer Chevrolet Twitter account retweeted it about 30min after the original post, which lead to post's sustained popularity. Tweet infectiousness against time as estimated by SEISMIC **(middle). Predictions of the tweet's final retweet count "Truth" as a function of time (bottom). We compare** SEISMIC **with Linear regression (LR). "Observed" plots the cummulative number of observed retweets by a given time.** SEISMIC **quickly finds an accurate estimate of the final retweet count.**

As the content gets stale the infectiousness may drop (see Figure 1), and thus, in SEISMIC we allow the infectiousness to freely vary over time. Moreover, our model is able to identify at each time point whether the cascade is in the *supercritical* or *subcritical* state, based on whether its infectiousness is above or below a critical threshold. Cascade in the supercritical state is going through a critical "explosion" period and its final size cannot be predicted accurately at the current time. However, when a cascade is in subcritical state it is tractable and we are able to predict its ultimate popularity accurately.

Our SEISMIC approach makes several contributions:

- **Generative model:** SEISMIC imposes no parametric assumptions, and requires no expensive feature engineering. Moreover, as complete social network structure may be hard to obtain, SEISMIC assumes minimal knowledge of the network: The only required input is the time history of reshares and the degrees of the resharing nodes.
- **Scalable and parallel computation:** Making a prediction using SEISMIC only requires computational time linear in the number of observed reshares. Since predictions for individual posts are made independently, SEISMIC can also be trivially parallelized.
- **Ease of interpretation:** For an individual cascade, the model synthesizes all its past history into a single infectiousness parameter. This infectiousness parameter holds a clear meaning, and can serve as input to other applications.

We evaluate our SEISMIC method on one month of complete Twitter data, where users posts tweets which others can then reshare by retweeting them. Such retweeting behavior can cause large information cascades to occur. We demonstrate that SEISMIC is able to predict the final retweet count of a given tweet with 30% better accuracy when compared to the state-of-the-art approach. For highly contagious tweets, our model achieves 15% relative error in predicting the final retweet count after observing the tweet for 1 hour, and 25% error after observing the tweet for just 10 minutes. Moreover, we also demonstrate how SEISMIC is able to identify tweets that will go "viral" and be among the most popular tweets in the future. By maintaining a shortlist of size 500 over time, we are able to cover 78 of the 100 most reshared tweets and 281 of the 500 most reshared tweets 10 minutes after they are posted.

The rest of the paper is organized as follows: Section 2 surveys the related work. Section 3 describes SEISMIC, and Section 4 shows how the model can be used to predict the final size of an information cascade. We evaluate our method and compare its performance with a number of baseline as well as state-of-the-art approaches in Section 5. And, in Section 6, we conclude and discuss future research directions.

2. RELATED WORK

The study of information cascades is a rich and fruitful field [26] and recent models for predicting size of information cascades can be characterized by two types of approaches, feature based methods and point process based methods.

Feature based methods first extract an exhaustive list of potentially relevant features, including content features, original poster features, network structural features and temporal features [6]. Then different learning algorithms are applied, such as simple regression models [2, 6], probabilistic collaborative filtering [34], regression trees [3], content-based models [23], and passive-aggressive algorithms [25]. There are several issues with such approaches: The performance is highly sensitive to the quality of the features [4, 29]. And, laborious feature engineering and extensive training are crucial for their success. Moreover, such approaches have limited applicability because they cannot be used in real-time online settings—given the massive amount of posts produced every second, it is practically impossible to extract all the necessary features for every post and then apply complicated prediction rules to each of them. In contrast, SEISMIC requires no feature engineering and results in an efficiently computable formula that allows SEISMIC to in real-time predict the final popularity of millions of posts as they are spreading through the network.

The second group of approaches are based on point processes, which directly model the formation of an information cascade in a network. Such models were mostly developed for the complementary problem of network inference, where one observes a number of information cascades and tries to infer the structure of the underlying network over which the cascades propagated [8, 10, 12, 14, 17, 32, 35]. These methods have been successfully applied to study the spread of memes on the web [10, 14, 31, 32] as well as hashtags

Symbol	Description
w	Post/information cascade
p_t	Infectiousness of w at time t (Section 3.2)
$\phi(s)$	Memory kernel (Section 3.1)
i	Node that contributed i^{th} reshare. $i = 0$ corresponds to the originator of the post.
t_i	Time of the i^{th} reshare relative to the original post.
n_i	Out-Degree of the i^{th} node
R_t	Cumulative popularity by time t: $\lvert\{i > 0; t_i \le t\}\rvert$
R_∞	Final popularity (final number of reshares): $\lvert\{i > 0\}\rvert$
N_t	Cumulative degree of resharers by time t: $\sum_{i:t_i \le t} n_i$
N_t^e	Effective cumulative degree of resharers by time t: $N_t^e = \sum_{i=0}^{R_t} n_i \int_{t_i}^{t} \phi(s - t_i) ds$
λ_t	Intensity of cumulative popularity R_t
$\hat{p}_t$	Model's estimate of infectiousness p_t at time t
$\hat{R}_\infty(t)$	Model's estimate at time t of final popularity R_∞

Table 1: Table of symbols.

on Twitter [35]. In contrast, our goal is not to infer an unobserved network but to predict the ultimate size of a cascade in an observed network.

A major distinction between our model and existing works based on Hawkes processes (*e.g.*, [21, 22, 32, 33, 35]) is that we assume the process intensity λ_t depends on another stochastic process p_t, the post infectiousness. In other words, we allow the infectiousness to change over time. Moreover, some of these methods [33] rely on computationally expensive Bayesian inference, while our method has linear time complexity. Another recently proposed related work is [13], which also takes the point process approach and directly aim to predict tweet popularity. However, their method makes restrictive parametric assumptions and does not consider the network structure, which limits its predictive ability. We compare SEISMIC with [13] in the experiment in Section 5.

3. MODELING INFORMATION CASCADES

Next we formally describe SEISMIC, a statistical model of information cascades. We discuss how SEISMIC can be used for:

1. Estimating spreading rate of a given information cascade, which we quantify by the post's infectiousness.
2. Determining whether the cascade is in supercritical (explosive) or subcritical (dying out) state.
3. Predicting the final size of the information cascade, which is measured by the ultimate number of reshares received by the post that stated the cascade.

Important quantities in our model are the post infectiousness p_t, the cascade speed of spreading λ_t, which is determined by human reaction time modeled by $\phi(s)$, and R_t which is the cumulative number of reshares of a given post up to time t. Our goal is to predict R_∞, which the ultimate number of reshares of a post.

We proceed by defining human reaction time model $\phi(s)$ in Section 3.1 and infectiousness p_t in Section 3.2. We then connect the two by describing our self-exciting model in Section 3.3. The model enables us to predict the final post popularity R_∞ at any point in time. Table 1 summarizes the notation.

3.1 Human reaction time

The first component of our model is human reaction time. In our problem of predicting post popularity, we need to know how long it takes for a person to reshare a post. We assume that the time between the appearance of a post in a users' timeline and a reshare of the same post by the user, denoted by s, is distributed with density $\phi(s)$. This is also called memory kernel in physics literature, because it measures a physical/social system's memory of stimuli [7].

The distribution of human response times $\phi(s)$ has been shown to be heavy-tailed in social networks [5]. Usually the tail of $\phi(s)$ is assumed to follow a power-law with exponent between 1 and 2 or a log-normal distribution [7, 33]. However, due to the rapid nature of information sharing on Twitter, it is also natural to expect many short-reaction times. In fact, our exploratory data analysis in Section 5.2 confirms that in Twitter, $\phi(s)$ is approximately constant for the first 5 minutes followed by a power-law decay. Different social networks usually have different distributions of human reaction times. However, $\phi(s)$ only needs to be estimated once for the whole network and thus we can safely assume it is given to us. Detailed estimation procedure of $\phi(s)$ can be found in Section 5.2.

3.2 Post infectiousness

The second component of our model is post infectiousness. We assume each post w is associated with a time dependent, intrinsic infectiousness parameter $p_t(w)$. In other words, $p_t(w)$ models how likely will the post w be reshared at time t. Infectiousness of a post may depend on an intricate combination of factors, including the quality of the post's content, poster's social network structure, current time of the day, poster's geographical location, and many others. However, as we assume no parametric form for p_t, our model is able to infer the right value of p_t, which accounts for all the factors, as well as how it evolves over time.

Literature studying self-exciting point processes mostly assumes p_t to be fixed over time. Consequently, an important concept is the *criticality* of the process R_t. In self-exciting point processes with constant infectiousness $p_t \equiv p$, there exists a phase transition phenomenon at certain critical threshold p^* such that [11]:

1. If $p > p^*$, then $R_t \to \infty$ as $t \to \infty$ almost surely and exponentially fast. This is called the *supercritical* regime.
2. If $p < p^*$, then $\sup_t R_t < \infty$ almost surely. This is called the *subcritical* regime.

In reality, R_t is always bounded due to the finite size of the network. Thus, if p_t is assumed to be a constant p, no supercritical cascades can exist. We propose a more flexible model of infectiousness that allows p_t to change over time. For example, as the post gets older, the information may get outdated and its spreading power (*i.e.*, infectiousness) might decrease. Similarly, as the post spreads further away from the original poster, its ability to spread (*i.e.*, infectiousness) might decay [33]. And, if a post is reshared by a highly influential user, that may increase the posts' infectiousness. Thus, rather than assuming a particular evolutionary pattern of p_t, we allow it to vary over time in a non-parametric way.

3.3 The SEISMIC model

To derive SEISMIC, we connect human reaction times and post infectiousness with the size of the information cascade, *i.e.*, number of reshares of a post. In order to link p_t to the post resharing process R_t, we assume R_t is actually a *doubly stochastic self-exciting point process*, an extension to the self-exciting point process also called the Hawkes process [15] and was initially used to model earthquakes [24].

We first define the intensity λ_t of R_t, which simply measures the rate of obtaining an additional reshare at time t. More formally:

$$\lambda_t = \lim_{\Delta \downarrow 0} \frac{\mathbb{P}\left(R_{t+\Delta} - R_t = 1\right)}{\Delta}.$$

In SEISMIC, the intensity λ_t at time t is determined by infectiousness p_t, reshare times t_i, node degrees n_i, and human reaction time distribution $\phi(s)$. The exact relationship described in Eq. (1) is inspired by the theory of Hawkes processes [15]:

$$\lambda_t = p_t \cdot \sum_{t_i \le t,\ i \ge 0} n_i \phi(t - t_i), \quad t \ge t_0. \tag{1}$$

Intuitively, $\sum_{t_i \le t,\ i \ge 0} n_i \phi(t - t_i)$ is the intensity of the arrival of newly exposed users at time t, so its product with the resharing probability at time t gives the intensity of reshares at time t.

Note that the above point process is *self-exciting* because each previous observation i ($t_i \le t$) contributes to the intensity λ_t. It is further *doubly stochastic* because the infectiousness p_t is itself a stochastic process.

Additionally, we assume $\{n_i\}$ are independent and identically distributed with some mean n^*. Mean degree n^* is related to the critical threshold p^* already discussed in Section 3.2. The critical infectiousness threshold takes value $p^* = 1/n^*$. We give the proof of this fact in Proposition 4.1.

4. PREDICTING INFORMATION CASCADES

Next we describe how to perform statistical inference for self-exciting model of cascades we just described in the previous section. Specifically, we discuss how SEISMIC estimates the infectiousness parameter p_t and then predicts the ultimate size of the cascade, that is, the total number of post's reshares R_∞.

Throughout this section, we assume the followers of all the resharers are disjoint. This assumption may not be true in general, however the conclusions made in this section remain valid if the node degree n_i is replaced by the "effective" node degree which is simply the total number of newly exposed followers, *i.e.*, the followers of i^{th} resharer who do not follow the first $i-1$ resharers or the original author of the post.

4.1 Estimating post infectiousness

We first define the *sample-function density*, which plays a central role in decision and estimation problems in a self-exciting point process [28]. Denote $\mathcal{F}_t = \{(n_i, t_i)\}_{i=0}^{R_t}$ as the information available by time t. The sequence of t_i is the increasing reshare occurrence times and n_i is the number of followers of the i^{th} resharer. Sample-function density is defined as the joint probability of the number of reshares in the time interval $[t_0, t)$ and the density of their occurrence times.

To motivate our estimator of p_t, we first consider the case that the infectiousness parameter remains constant over time, *i.e.*, $p_t \equiv p$. Later we will relax this assumption and allow p_t to vary over time.

In SEISMIC, the sample-function density can be expressed using the intensity λ_t as [28, Thm. 6.2.2]

$$\mathbb{P}(R_t = r, t_1, \ldots, t_r) = \prod_{i=1}^{R_t} \lambda_{t_i} \cdot \exp\left\{-\int_{t_0}^{t} \lambda_s ds\right\}. \tag{2}$$

By taking derivative of the log of Eq. (2) and combining it with Eq. (1), we obtain the maximum likelihood estimate (MLE) of $p(t)$:

$$\hat{p}(t) = \frac{R_t}{\sum_{i=0}^{R_t} n_i \int_{t_i}^{t} \phi(s - t_i) ds} \triangleq \frac{R_t}{N_t^e} \tag{3}$$

The above equation forms the basis of SEISMIC as it allows us to estimate the infectiousness $\hat{p}(t)$ at any given time t. Moreover, a confidence interval of $p(t)$ can also be obtained [28]. Notice that in the calculations above, we implicitly assume node degrees $n_0, n_1, \ldots$ are given and related to the process through Eq. (1).

In the estimator in Eq. (3), the numerator R_t, is the current number of reshares of a given post and the denominator N_t^e can be thought as the accumulative "effective" number of exposed users to the post. To shed more light on our estimator, we take $t \to \infty$, which leads to:

$$\hat{p}(\infty) = \frac{1}{\frac{1}{R_\infty} \sum_{j=0}^{R_\infty} n_j} \approx \frac{1}{n_*} \tag{4}$$

This means if we assume the infectiousness p_t is a constant over time, most of the posts will have the same infectiousness. This cannot explain the bursty and volatile dynamics observed only for some posts. Note that the above approximation is true for most of the realizations such that R_∞ is large. If a post was created by a high-degree node, then n_0 is very large, and hence $\hat{p}(\infty)$ could in fact be smaller than $1/n_*$.

The observation Eq. (4) as well as tweets like the one in Figure 1 motivate us to allow p_t varying over time. In this case, we smooth the MLE in Eq. (3) by only using observations close to time t to estimate p_t. In particular, we rely on a sequence of one-sided weighting kernels $K_t(s)$, $s > 0$, indexed by time t and estimate post infectiousness:

$$\begin{aligned} \hat{p}(t) &= \frac{\int_{t_0}^{t} K_t(t-s) dR_s}{\int_{t_0}^{t} K_t(t-s) dN_s^e} \\ &= \frac{\sum_{i=1}^{R_t} K_t(t - t_i)}{\sum_{i=0}^{R_t} n_i \int_{t_i}^{t} K_t(t-s)\phi(s-t_i) ds}. \end{aligned} \tag{5}$$

Notice that when $K_t(s) \equiv 1$ the estimator reduces to the MLE we derived in Eq. (3). In SEISMIC we use a triangular kernel with growing window size $t/2$ as weighting kernel $K_t(s)$:

$$K_t(s) = \max\left\{1 - \frac{2s}{t}, 0\right\}, \quad s > 0. \tag{6}$$

We chose the triangular kernel because it has properties important for our application. First, the kernel discards all posts that are older than $t/2$. In particular, it quickly discards the unstable and possibly explosive period at the beginning, which if included, would introduce an upward bias to p_t. Second, the kernel takes into account posts in a larger window size as time t increases. According to our experiments, the growing window size helps to stabilize $\hat{p}(t)$ compared to a fixed window size. Third, for reshares within the window, the kernel up-weights the most recent ones and gradually down-weights older ones. This keeps our estimator $\hat{p}(t)$ closer to the ever-changing true p_t. And last, since $K_t(s)$ is piece-wise linear, the integral $\int K_t(t-s)\phi(s-t_i)ds$ has a closed form for many different functions $\phi(s)$, including a constant followed by a power-law decay, which is the $\phi(s)$ we use in SEISMIC (Eq. (9)).

4.2 Predicting final popularity

Having described the procedure for inferring the post infectiousness, we now need to account for the network structure in order to predict how the post is going to spread across the network. Following, we can predict post's *final* reshare count at any point in post's lifetime.

For simplicity, let us assume the post is first posted at time 0, *i.e.*, $t_0 = 0$. Consider we have observed the post for t time units and our goal now is to predict post's final reshare count, R_∞, based on the information we have observed so far, $\mathcal{F}_t$.

The following proposition shows how to compute the expected final reshare count of a post. The main idea is to model an information cascade spreading over the network with a branching process

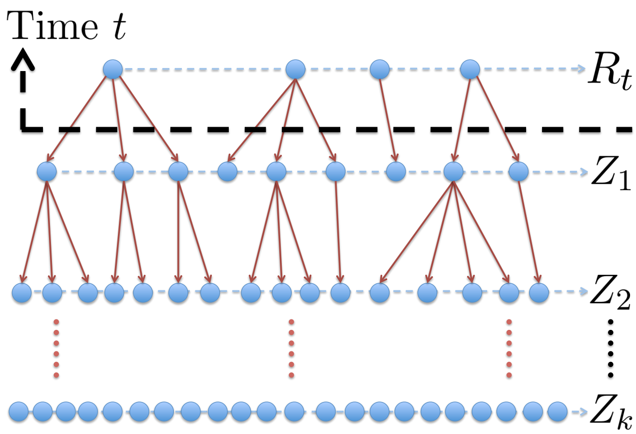

Figure 2: An illustration of the information diffusion tree after time t. Z_k denotes the number of reshares caused by the k^{th} generation descendants. Note that the final reshare count R_∞ is simply $R_t + \sum_{k=1}^{\infty} Z_k$.

that counts how posts are reshared, as illustrated in Figure 2. Predictor for R_∞ used by SEISMIC can be stated as follows:

PROPOSITION 4.1. *Assume the (out-)degrees in the network are i.i.d. with expectation n_* and the infectiousness parameter p_s is a constant p for $s \geq t$. Then, we have*

$$\mathbb{E}[R_\infty | \mathcal{F}_t] = \begin{cases} R_t + \dfrac{p(N_t - N_t^e)}{1 - pn_*}, & \text{if } p < \dfrac{1}{n_*}, \\ \infty, & \text{if } p \geq \dfrac{1}{n_*}. \end{cases} \tag{7}$$

PROOF. We derive Eq. (7), which allows us to predict the expected final reshare count, as follows. First, we consider the case where $p < 1/n_*$. We define the sequence of auxiliary random variables, $\{Z_1, Z_2, Z_3, \dots\}$ that aim to model the future shape of the information diffusion tree as illustrated in Figure 2. Our intuition is as follows: We have just observed the cascade (all the post reshares R_t) by time t and now we aim to model how the cascade will further spread throughout the network. We model this as a branching process, where Z_k denotes the number of reshares made by the k^{th} generation descendants (counting from generation R_t onwards). Thus, 1st generation descendants Z_1 refers to the number of new reshares generated by the posts created before time t, 2nd generation descendants Z_2 refers to the reshares of the posts of the 1st descendants, and so on (Figure 2). Notice that the summation over the Z_k's gives the post's final reshare count. In the following we use descendants Z_k only for derivation and emphasize that our final estimator does not require explicit descendant information.

Given Z_1, the sequence of random variables Z_k defines a Galton-Watson tree with the *offspring expectation* $\mu = n_* p$ [11]. Here, μ denotes the expected number of reshares that user's post gets. Using the standard branching process result that Z_i/μ^i is a martingale, we obtain that $\forall i > 1$,

$$\mathbb{E}\left[Z_{k+1} | Z_k\right] = \mu \, Z_k,$$

and therefore,

$$\mathbb{E}\left[\sum_{k=1}^{\infty} Z_k \middle| Z_1\right] = \frac{Z_1}{(1-\mu)} = \frac{Z_1}{(1-n_* p)}.$$

Hence, we obtain

$$\mathbb{E}[R_\infty | \mathcal{F}_t] = R_t + \mathbb{E}\left[\sum_{k=1}^{\infty} Z_k\right] = R_t + \frac{\mathbb{E}[Z_1]}{(1-n_* p)},$$

which ends up being the right hand side in Eq. (7) because $\mathbb{E}[Z_1] = p(N_t - N_t^e)$ by the definition of Z_1 and N_t^e.

Next, consider the case where $p = \hat{p}_t \geq 1/n_*$. In this regime, the point process is supercritical and stays explosive. In terms of the Galton-Watson tree discussed above, the offspring expectation $\mu = n * p \geq 1$, so $\mathbb{E}[Z_{k+1}] \geq \mathbb{E}[Z_k] \geq \cdots \geq \mathbb{E}[Z_1]$. So, the total future reshares $\sum_{k=1}^{\infty} Z_k$ has infinite expectation and the final reshare count cannot be reliably predicted. □

Note that the Proposition 4.1 assumes that $p_s = p_t$ for $s \geq t$, which may sometimes be unrealistic. To accommodate for this, we slightly change the prediction formula in Eq. (7) by adding two scaling constants α_t, γ_t that adjust the final prediction:

$$\hat{R}_\infty(t) = R_t + \alpha_t \frac{\hat{p}_t(N_t - N_t^e)}{1 - \gamma_t \hat{p}_t n_*}, \; 0 < \alpha_t, \gamma_t < 1 \,. \tag{8}$$

Our intuition to introduce these correction factors is the following. We expect α_t to decrease over time t and this ways scale-down the estimated infectiousness to account for the post getting stale and outdated. Similarly, γ_t accounts for the overlap in the neighborhoods of reposters' followers and thus over time as the post spreads farther in the network, we expect γ_t to increase as more nodes get exposed multiple times, and thus the arrival rate of new previously unexposed nodes decreases over time.

We use same values of α_t and γ_t for all posts and may change over time. The values of α_t and γ_t are selected to minimized median Absolute Percentage Error (refer to Section 5.4 for definition) on the training set. As described in Section 5.2, we find α_t is more important than γ_t in practice.

Algorithm 1 SEISMIC: Predict final cascade size

Purpose: For a given post at time t, predict its final reshare count
Input:

- **Post resharing information:** t_i and n_i for $i = 0, \dots, R_t$

Algorithm:
$N_t = 0$, $N_t^e = 0$
for $i = 0, \dots, R_t$ **do**
  N_t += n_i
  N_t^e += $n_i \int_{t_i}^{t} \phi(s - t_i) ds$ (Sec. 3.1)
end for
$\hat{R}_\infty(t) = R_t + \alpha_t \hat{p}_t (N_t - N_t^e)/(1 - \gamma_t \hat{p}_t n_*)$ (Alg. 2)
Deliver: $\hat{R}_\infty(t)$

4.3 The SEISMIC algorithm

Last, we put together all the components described so far and synthesize them in the SEISMIC algorithm. SEISMIC algorithm for predicting $\hat{R}_\infty(t)$ is described in Algorithm 1, which uses the algorithm for computing $\hat{p}_t$ (Algorithm 2) as a subroutine. The algorithms are based on Eqs. (5) and (8). We assume parameters $K_t(s)$, α_t, γ_t, n_* are given *a priori* or estimated from the data.

Computational complexity of SEISMIC. For any choice of $\phi(s)$ and $K_t(s)$, the computational cost of SEISMIC is $O(R_t)$ for both calculating $\hat{p}_t$ and predicting $\hat{R}_\infty(t)$. Of course, the actual computing time depends heavily on the computation of the integral $\int_{t_i}^{t} K_t(t-s)\phi(s-t_i)ds$ and $\int_{t_i}^{t} \phi(s-t_i)ds$, however, overall,

Algorithm 2 Compute real-time infectiousness $\hat{p}(t)$

Purpose: For a given post w, calculate infectiousness p_t with information about w prior to time t
Input:

- **Post resharing information:** t_i and n_i for $i = 0, \ldots, R_t$

Algorithm:
$\tilde{R}_t = 0$, $\tilde{N}_t^e = 0$
for $i = 0, \ldots, R_t$ **do**
  $\tilde{R}_t$ += $K_t(t - t_i)$
end for
for $i = 0, \ldots, R_t$ **do**
  $\tilde{N}_t^e$ += $n_i \int_{t_i}^{t} K_t(t-s)\phi(s-t_i)ds$ (Sec. 4.1)
end for
$p_t = \tilde{R}_t / \tilde{N}_t^e$
Deliver: p_t

the computational cost of SEISMIC is *linear* in the observed number of reshares R_t of a given post by time t.

The linear time complexity is in part also due to the shape of our memory kernel. In Section 5.2 we will estimate the memory kernel $\phi(s)$ for Twitter to have the following form (for some $s_0 > 0$):

$$\phi(s) = \begin{cases} c & \text{if } 0 < s \leq s_0, \\ c(s/s_0)^{-(1+\theta)} & \text{if } s > s_0. \end{cases} \tag{9}$$

This means that with the memory kernel $\phi(s)$ in Eq. (9) and triangular weighting kernel $K_t(s)$ in Eq. (6), both integrals can be evaluated in closed form because they are piecewise polynomials (polynomial with possibly non-integer exponents). We omit the details for brevity.

5. EXPERIMENTS

In this section, we describe the Twitter data set, our parameter estimation procedure, and compare the performance of SEISMIC to baselines as well as state-of-the-art estimators.

5.1 Data description and data processing

Our data is the complete set of over 3.2 billion tweets and retweets on Twitter from October 7 to November 7, 2011. For each retweet, the data provides information on the original tweet id, original post time, retweet time, and number of followers of the retweeter.

We focus on a subset of tweets with at least 50 retweets, so that our model enables the prediction as soon as sufficient number of retweets occurs. There are 166076 tweets satisfying this criterion in the first 15 days. We form the training set using the tweets from the first 7 days and the test set using the tweets from the next 8 days. We use the remaining 14 days for the retweet cascades to unfold and evolve. For a particular retweet cascade, we obtain all the retweets posted within 14 days of the original post time, *i.e.*, we approximate R_∞ by $R_{14\text{ days}}$. We estimate parameters $\phi(s), \alpha_t, \gamma_t$ and n_* with the training set, and evaluate the performance of the estimator $\hat{R}_\infty$ on the test set. For the tweets in our training set, $R_{14\text{ days}}$ has mean 209.8 and median 110. The temporal evolution of mean and median of R_t are also shown in Figure 3.

5.2 SEISMIC parameter estimation

Next, we describe the fitting of $\phi(s), \alpha_t, \gamma_t$ and n_*. First we fit the memory kernel $\phi(s)$ with the training set. We take 15 subcritical tweets, and assume all their retweets come from immediate followers. Under this assumption, the reaction time (Section 3.1) is the same as the relative retweet time.

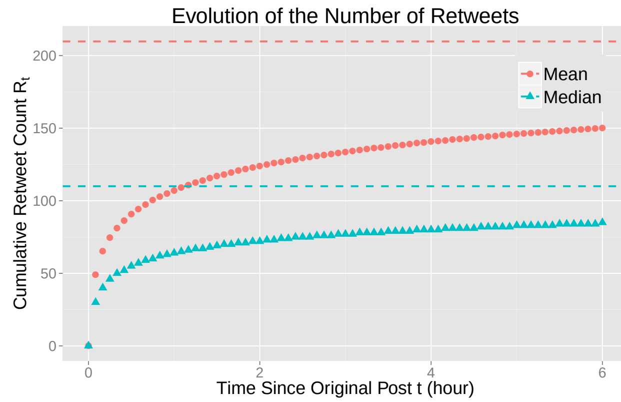

Figure 3: The red dotted line shows the evolution of the mean of cumulative retweet count R_t, while the green triangular line shows the median. The dashed horizontal lines correspond to mean and median final retweet count $R_{14\text{ days}}$.

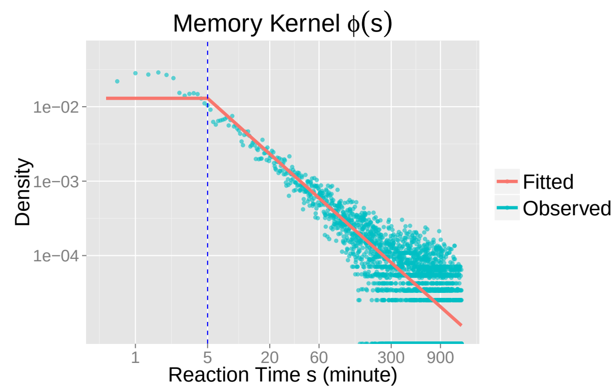

Figure 4: Plot of observed reaction time distribution and estimated memory kernel $\phi(s)$. The reaction time is plotted on a log scale, hence a linear trend in the plot suggests a power law decay in the distribution.

The observed reaction time distribution plotted in Figure 4 suggests a form of Eq. 9 for the memory kernel: constant in the first 5 minutes, followed by a power-law decay. After setting the constant period s_0 to 5 minutes, we estimated power law decay parameter $\theta = 0.242$ with the complimentary cumulative distribution function (ccdf), and chose $c = 6.265725 \times 10^{-4}$ to make $\int_0^\infty \phi(s)ds = 1$. The memory kernel is a network wide parameter, hence only needs to be estimated once. The fitted memory kernel is plotted in Figure 4.

Last, we briefly comment on the correction factors α_t and γ_t introduced in Eq. (7). We use the same values of α_t and γ_t for all tweets. Notice that γ_t and n_* only affect the predictions through their product $\gamma_t n_*$. Overall, we find the value of $\gamma_t n_*$ has little effect on the performance of our algorithm. In our experiments we simply set $\gamma_t n_* = 20$ for all t. We choose the value of α_t such that it minimizes the training median Absolute Percentage Error (Section 5.4). We report values of α_t in Table 2. α_t has a particularly small value at $t = 5$ minutes, which may be a result of the overestimation of p_t, when the triangular kernel has not moved away from the unstable beginning period. After that α_t begins a slow and consistent decay to account for the fact that information is getting increasing stale and outdated over time.

Now, we are ready to apply SEISMIC (Algorithms 2 and 1) and for a given tweet w for every 5 minute interval t output our estimate $\hat{R}_\infty(t, w)$ of the final retweet count $R_\infty(w)$.

time (minute)	5	10	15	20	30
α	0.389	0.803	0.772	0.709	0.680
time (minute)	60	120	180	240	360
α	0.562	0.454	0.378	0.352	0.326

Table 2: List of α_t used in Algorithm 1.

5.3 Baselines for comparison

We consider four baseline estimators for comparison. The first two are regression based and the next two are point process based.

- **Linear regression (LR) [30]:** The model can be defined as

$$\log R_\infty = \alpha_t + \log R_t + \epsilon,$$

where ϵ denotes the Gaussian noise. This is also the second baseline estimator used in [33]. Notice that all the tweets receive the same multiplicative constant for a given time.

- **Linear regression with degree (LR-D) [30]:** This model can be written as

$$\log R_\infty = \alpha_t + \beta_{1,t} \log R_t + \beta_{2,t} \log N_t + \beta_{3,t} \log n_0 + \epsilon$$

where ϵ denotes, as before, the Gaussian noise. LR-D is more flexible than LR, since it allows $\log R_t$ has slope not equal to 1 and uses additional features such as the cumulative degree and the tweet originator's degree.

- **Dynamic Poisson Model (DPM) [2, 7]:** It models the retweet times $\{t_k\}$ as a point process with rate

$$\lambda_t = \lambda_{t_{\text{peak}}}(t - t_{\text{peak}})^\gamma$$

where $t_{\text{peak}} = \arg\max_{s<t} \lambda_s$. The power-law parameter γ is estimated separately for each tweet. To discretize the model, we bin time into $b = 5$ minutes intervals. Note that when $\gamma > -1$, the integral $\int_{t_{\text{peak}}+b}^{\infty} \lambda_t dt$ is infinite. If this is the case, we move t_{peak} forward to the second to maximum bin.

- **Reinforced Poisson Model (RPM) [27]:** This recently published state-of-the-art approach models rate of retweet times as

$$\lambda_t = c f_\gamma(t) r_\alpha(R_t)$$

where parameter c measures the attractiveness of the message, $f_\gamma(t) \propto t^{-\gamma} (\gamma > 0)$ models the aging effect, and $r_\alpha(R_t)(\alpha > 0)$ is the reinforcement function which depicts the "rich get richer" phenomenon. Given a particular tweet, the parameters c, γ, α are found by maximizing the likelihood function, where the optimal values are projected to their feasible sets whenever they are outside.

5.4 Evaluation metrics

We use the following comprehensive evaluation metrics to compare our method with the baseline estimators. For a particular tweet, suppose that the prediction for R_∞ at time t is denoted by $\hat{R}_\infty(t)$. We use the following criteria for evaluation:

- **Absolute Percentage Error (APE):** For a given tweet w and a prediction time t, the APE metric is defined as,

$$\text{APE}(w, t) = \frac{|\hat{R}_\infty(w, t) - R_\infty(w)|}{R_\infty(w)}.$$

When APE metric is used for evaluation purposes, various quantiles of APE over the tweets (all possible w) in the test dataset will be reported at each time t.

- **Kendall-τ Rank Correlation:** This is a measure of rank correlation [18], which computes the correlation between the ranks of $\hat{R}_\infty(t)$ and R_∞ for all test tweets. This metric is generally more robust than Pearson's correlation between the exact values of $\hat{R}_\infty(t)$ and R_∞. A high value of rank correlation means our predictoin and the final retweet counts are strongly correlated.
- **Breakout Tweet Coverage:** We create a ground-truth list of top-k tweets with the highest final retweet count. We refer to these tweets as "breakout" tweets. Using our model we can also produce a top-k list based on the predicted final retweet count. We evaluate the two lists by comparing how well does the predicted top-k cover the ground-truth top-k list. We give additional motivation and details for this metric in Section 5.5.3.

5.5 Experimental results

In this section, we evaluate the performance of our SEISMIC method and the four baselines described in Section 5.3. All the methods start making predictions as soon as a given tweet gets retweeted 50 times.

5.5.1 SEISMIC model validation

First, we aim to empirically validate the SEISMIC prediction question and the claim in Proposition 4.1. In Proposition 4.1, we obtained a formula of the expected number of final retweets in terms of the infectiousness parameter p_t. Our goal is to show that Proposition 4.1 actually gives an unbiased estimate of the true final retweet count. We proceed as follows.

We use SEISMIC to make a prediction after observing each tweet for 1 hour and then plot the prediction against the true final number of retweets. If SEISMIC gives an unbiased estimate, then we expect a linear curve $y = x$, that is, the expected predicted $\hat{R}_\infty$ matches the true expected R_∞.

Figure 5 shows that the empirical average almost perfectly coincides with SEISMIC's prediction. This suggests that SEISMIC estimator in Eq. (7) is unbiased and we can safely use it to predict the expected final number of retweets.

However, as mentioned earlier after Proposition 4.1, in practice one often wants to shrink the prediction in order to stabilize the estimator and achieve better performance. This is particularly important if the error metric is Absolute Percentage Error (APE), since an underestimation can at most result in 100% error but an overestimation could incur a much worse error. Therefore, we use the calibrated prediction formula Eq. (8) for the rest of the experiments in this section. Note, that we apply the similar calibration procedure for all baselines as well, as we found that it significantly improves their performance as well.

5.5.2 Predicting final retweet count

We run our SEISMIC method for each tweet and compute the Absolute Percentage Error (APE) metric as a function of time. We plot the quantiles of the distribution of APE of SEISMIC in Figure 6. After observing the cascade for 10 minutes (t = 10min), the 95th, 75th, and 50th percentiles of APE are less than 71% , 44%, and 25%, respectively. This means that after 10 minutes, average error is less than 25% for 50% of the tweets and less than 71% for 95% of them. After 1 hour the error gets even lower—APE for 95%, 75% and 50% of the tweets drops to 62%, 30% and 15%, respectively.

The proposed method, SEISMIC, demonstrates a clear improve-

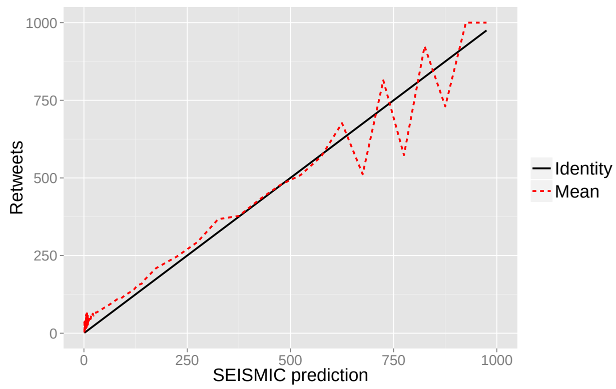

Figure 5: An empirical validation of our SEISMIC model. The empirical average of 14 days retweets (red dotted line) nicely follows the SEISMIC prediction (black solid line).

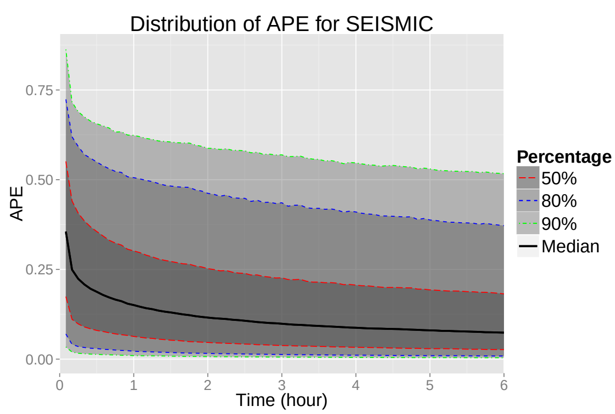

Figure 6: Absolute Percentage Error (APE) of SEISMIC on the test set. We plot the median and the middle $50\%, 80\%, 90\%$ percentiles of the distribution of APE across the tweets.

ment over the baselines as shown by Figures 7 and 8. The left panel of Figure 7 and 8 shows the median Absolute Percentage Error (APE) of different methods over time as more and more of the retweet cascade gets revealed. The Linear Regression model (LR) and Linear Regression model with Degree (LR-D) have very similar performance, indicating the additional features used by LR-D are not very informative. DPM performs poorly across the entire tweet lifetime, while the other point process baselines RPM is worse than LR and LR-D in the early period but becomes better after about 2 hours. Overall, SEISMIC is about 30% more accurate than all the baselines across the entire tweet lifetime in median APE.

Similarly, the right panels of Figures 7 and 8 show the Kendall-τ rank correlation between the predicted ranking of top most retweeted tweets and the truth ranking of tweets by their ultimate retweet count. Again SEISMIC is giving much more accurate rankings than the baselines.

5.5.3 *Identifying breakout tweets*

Can we identify a breakout tweet before it receives most of its retweets? This question arises from various applications like trend forecasting or rumor detection. The goal of this prediction task is

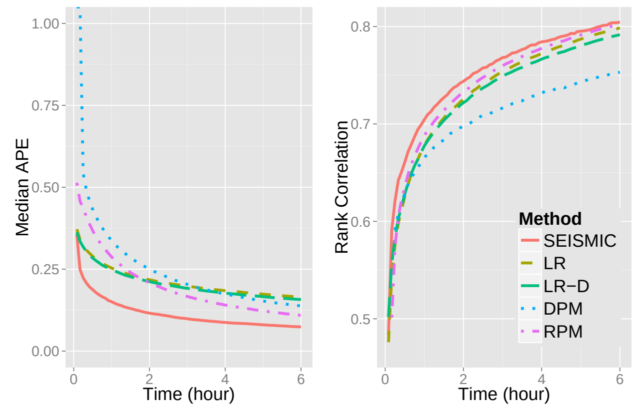

Figure 7: Median Absolute Percentage Error (APE) and Kendall's Rank Correlation of SEISMIC and the baselines as a function of time. SEISMIC consistently gives best performance.

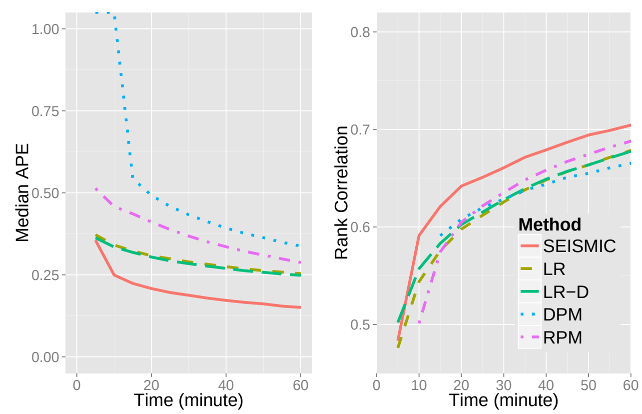

Figure 8: Zoom-in of Figure 7: Median APE and Rank Correlation for the first 60 minutes after the tweet was posted. SEISMIC performs especially well early in tweet's lifetime.

to as early as possible identify "breakout" tweets, which have the highest final retweet count. We compare performance of different models in detecting breakout tweets using models' predictions of tweet's final retweet count.

First, we form a ground-truth set L_M^* of size M. Set L_M^* contains top-M tweets with the highest final retweet count. Then with each of the prediction methods, we produce a sequence of size m lists, $\hat{L}_m(t)$. At each time t list $\hat{L}_m(t)$ contains the top-m tweets with the highest predicted retweet count.

As described in Section 5.4, we then compare each $\hat{L}_m(t)$ with L_M^*, and calculate the *Breakout Tweet Coverage*, which is defined as the proportion of tweets in L_M^* covered by $\hat{L}_m(t)$.

Figure 9 shows the performance of SEISMIC for detecting top 100 most retweeted tweets (L_{100}^*) as a function of time. SEISMIC is able to cover 82 tweets in the first 1 hour and 93 tweets in the first 6 hours.

The 5th most retweeted tweet in this plot is actually the tweet [1] we showed earlier in Figure 1. Notice how SEISMIC detects this tweet 30 minutes after it was posted, while LR and LR-D take more than an hour. DPM fails to detect this breakout for the first 6 hours because the power constant γ is never less than -1.

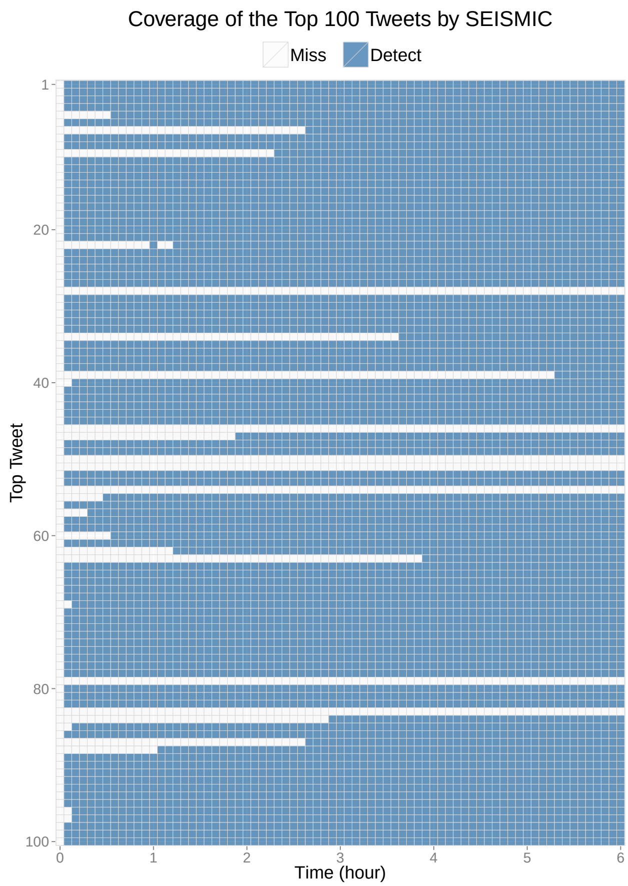

Figure 9: Time changing coverage of the top 100 tweets by shortlist of 500 tweets generated from our SEISMIC model. Rows i represent tweets with ith largest final total retweet count. White blocks in row i and time t indicate tweet i was not covered by our predicted list of top-500 tweets at time t, and blue indicates successful coverage.

To compare SEISMIC with baseline methods, we keep the size of the predicted lists to be $m = 500$, and use a larger target list L^*_{500}, which is a more difficult task than finding L^*_{100}. Figure 10 compares the coverage of different methods against the proportion of retweets seen. After seeing 20% of the retweets of these 500 tweets, SEISMIC covers 65% of them in the shortlist, while LR-D and LR both cover 50%. In general, the dynamic Poisson models fail to provide accurate predictions and breakout identifications.

Overall, SEISMIC allows for effective detection of breakout tweets. For instance, after seeing around 25% of the total number of retweets of a given tweet (in other words, after observing a tweet for around 5 minutes), SEISMIC can identify 60% of the top-100 tweets according to the final retweet count.

5.6 Discussion of model robustness

SEISMIC demonstrates better robustness than the other two point process based methods — DPM and RPM. While SEISMIC doesn't predict for supercritical tweets, DPM and RPM fails to predict when the parameter is in the infeasible set ($\gamma < -1$ for DPM and $\gamma < 0$ or $\alpha < 0$ for RPM). For example, in Figure 1, SEISMIC characterizes the tweet as supercritical for the first 70 min, DPM fails to predict for the first 6 hours and RPM can only predict from 30 to 80 minutes. On average, for all tweets with at least 50 retweets, when making predictions at 15 minutes, 1 hour and 6 hours, SEISMIC considers 1.80%, 1.29% and 0.67% of the tweets to be supercritical. In comparison, DPM fails to make predict for 6.77%, 5.79% and 1.45%, and RPM fails for 3.45%, 5.69% and 15.43% of the tweets.

Our SEISMIC method is also significantly faster than the RPM model [27], which requires to solve a nonlinear optimization problem every time it predicts. In our implementation, the average running time per tweet for predicting at every 5 minutes for 6 hours is 0.02s for SEISMIC and 3.6s for RPM. The reported running time includes both parameter learning and prediction.

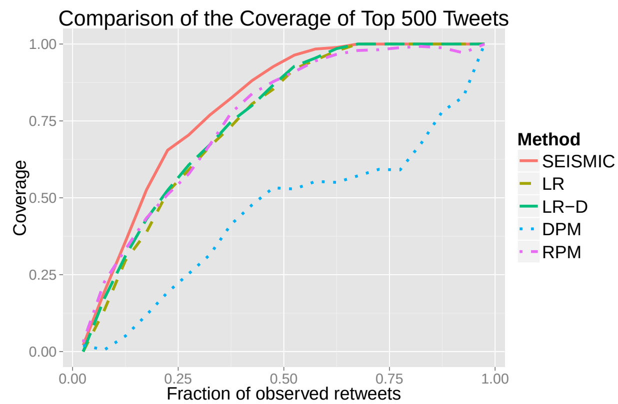

Figure 10: Comparison of coverage of the target list L^*_{500} by different methods. SEISMIC starts to have a real advantage over all baseline methods after about 10% of retweets of a given tweet are seen. All methods except for DPM have perfect coverage after about 65% of retweets of a given tweet are seen.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed SEISMIC, a flexible framework for modeling information cascades and predicting the final size of an information cascade. Our method differs from others in the following aspects:

- We model the information cascades as self-exciting point processes on Galton-Watson trees. Our SEISMIC approach provides a theoretical framework for explaining temporal patterns of information cascades.
- SEISMIC is both scalable and accurate. The model requires no feature engineering and scales linearly with the number of observed reshares of a given post. SEISMIC provides a way to predict information spread for millions of posts in an online real-time setting.
- SEISMIC brings extra flexibility to estimation and prediction tasks as it requires minimal knowledge about the information cascade as well as the underlying social network structure.

Last, we also mention that, there are many interesting venues for future work and our proposed model can be extended in many different directions. For example, if network structure is available, one could replace the node degree n_i by the number of newly exposed followers. If content-based features or features of the original post are available, one could easily formulate a prior of p_t for each post. If temporal features such as the user's time zone are available, one could directly use them to modify the estimator $\hat{p}_t$. In this sense, the proposed model provides an extensible framework for predicting information cascades.

Overall, we presented a statistically sound and scalable bottom-up model of information cascades that allows for predicting final cascade size as the cascade unfolds over the network. We hope that our framework will prove useful for developing richer understanding of cascading behaviors in online networks and will pave ways towards better management of shared content and applications that can identify trending content early.

Data and Software

We will release the dataset and a software of SEISMIC upon publication.

Acknowledgements

The authors would like to thank David O. Siegmund for his constructive suggestions.

7. REFERENCES

[1] https://twitter.com/mottbollomy/status/127001313513967616.

[2] D. Agarwal, B.-C. Chen, and P. Elango. Spatio-temporal models for estimating click-through rate. WWW '09, 2009.

[3] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. WSDM' 11.

[4] R. Bandari, S. Asur, and B. A. Huberman. The pulse of news in social media: Forecasting popularity. In *ICWSM '12*, pages 26–33, 2012.

[5] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[6] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? WWW '14.

[7] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41), 2008.

[8] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. ICML '14, 2014.

[9] P. A. Dow, L. A. Adamic, and A. Friggeri. The anatomy of large facebook cascades. In *ICWSM '13*, 2013.

[10] N. Du, L. Song, M. Yuan, and A. J. Smola. Learning networks of heterogeneous influence. NIPS '12, 2012.

[11] R. Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

[12] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. NIPS '14, 2014.

[13] S. Gao, J. Ma, and Z. Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *WSDM '15*, 2015.

[14] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML*, 2013.

[15] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1), 1971.

[16] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *WWW '11*, 2011.

[17] D. Hunter, P. Smyth, D. Q. Vu, and A. U. Asuncion. Dynamic egocentric models for citation networks. In *ICML '11*, 2011.

[18] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[19] A. Kupavskii, L. Ostroumova, A. Umnov, S. Usachev, P. Serdyukov, G. Gusev, and A. Kustarev. Prediction of retweet cascade size over time. In *CIKM '12*, 2012.

[20] D. Liben-Nowell and J. Kleinberg. Tracing the flow of information on a global scale using Internet chain-letter data. *Proceedings of the National Academy of Sciences*, 105(12):4633–4638, 2008.

[21] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: Model and implications. KDD '12, 2012.

[22] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

[23] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *ACM WebSci '11*, 2011.

[24] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83, 1988.

[25] S. Petrovic, M. Osborne, and V. Lavrenko. RT to Win! Predicting Message Propagation in Twitter. In *ICWSM '11*, 2011.

[26] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[27] H.-W. Shen, D. Wang, C. Song, and A.-L. Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. *arXiv preprint arXiv:1401.0778*, 2014.

[28] D. L. Snyder and M. I. Miller. *Random Point Processes in Time and Space*. Springer, 2011.

[29] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *SOCIALCOM '10*, 2010.

[30] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, Aug. 2010.

[31] J. Yang and J. Leskovec. Patterns of temporal variation in online media. WSDM '11, 2011.

[32] S.-H. Yang and H. Zha. Mixture of mutually exciting processes for viral diffusion. In *ICML '13*, 2013.

[33] T. Zaman, E. Fox, and E. Bradlow. A Bayesian approach for predicting the popularity of tweets. *Annals of Applied Statistics*, 8(3):1583–1611, 2014.

[34] T. R. Zaman, R. Herbrich, J. Van Gael, and D. Stern. Predicting information spreading in twitter. In *Computational Social Science and the Wisdom of Crowds Workshop, NIPS*, 2010.

[35] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS '13*, pages 641–649, 2013.