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 allow users to create and share content. Big information cascades of post resharing can form as users of these sites reshare others' posts with their friends and followers. 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 make accurate predictions. Our model requires no training or expensive feature engineering. It results in a simple and efficiently computable formula that allows us to answer questions, in real-time, such as: Given a post's resharing history so far, what is our current estimate of its final number of reshares? Is the post resharing cascade past the initial stage of explosive growth? And, which posts will be the most reshared in the future?

We validate our model using 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 average 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: information diffusion; cascade prediction; self-exciting point process; contagion; social media.

1. INTRODUCTION

Online social networking services, such as Facebook, Youtube, and Twitter, allow their users to post and share content in the form of posts, images, and videos [9, 17, 21, 30]. 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. This way large information cascades of post resharing spread through the network.

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

Predicting the ultimate popularity of a post 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. Accurate prediction would allow Twitter to rank content better, discover trending posts faster, and improve its content-delivery networks. Moreover, predicting information cascades allows us to gain fundamental insights into predictability of collective behaviors where uncoordinated actions of many individuals lead to 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 future growth [4, 6, 17, 20, 26, 30]. However, feature extraction can be expensive and cumbersome, and one is never sure if more effective features could be extracted. The question remains how to design a simple and principled bottom-up model of cascading behavior. The challenge lies in defining a model for an individual's behavior and then aggregating the effects of the individuals in order to make an accurate global prediction.

Present work. Here we focus on predicting the final size of an information cascade spreading through a network. We develop a statistical model based on the theory of *self-exciting point processes*. A point process indexed by time is called a *counting process* when it counts the number of instances (reshares, in our case) over time. In contrast to homogeneous Poisson processes which assume constant intensity 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" phenomena [22, 23, 33, 36]. They are ideal for modeling information cascades in networks because every new reshare of a post not only increases its cumulative reshare count by one, 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 measures the reshare probability. We allow the infectiousness to vary freely over time in agreement with the observation that the infectiousness can drop as the content gets stale (see Figure 1).

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783401.

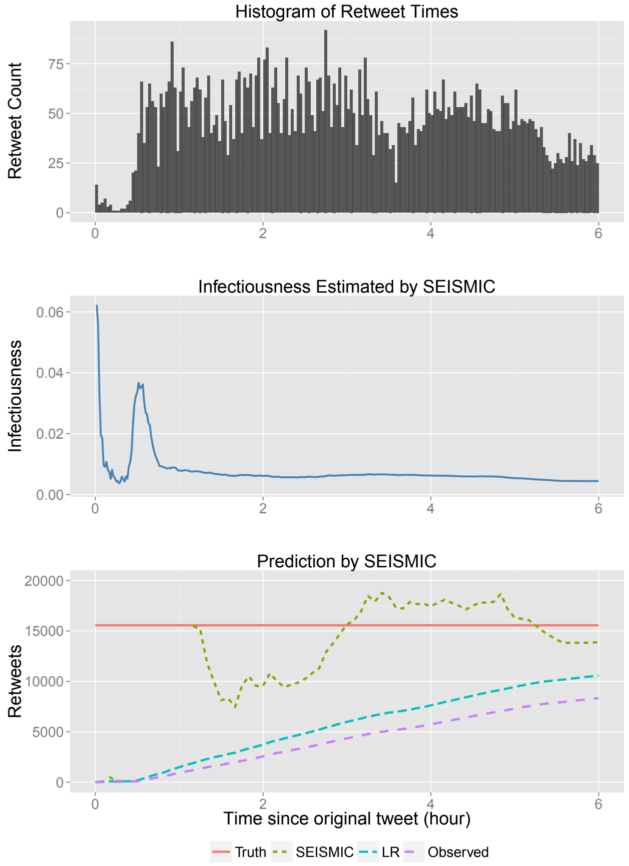

Figure 1: First 6 hours of retweeting activity of a popular tweet [1] (top). The controversial tweet is about the fresh death of dictator Muammar Gaddafi and mentions singer Justin Bieber. Interestingly, the car manufacturer Chevrolet Twitter account inappropriately retweeted the tweet about 30 minutes after the original tweet, which possibly lead to tweet's sustained popularity. Tweet infectiousness against time as estimated by SEISMIC (middle). Predictions of the tweet's final retweet count (denoted as "Truth") as a function of time (bottom). We compare SEISMIC with time series linear regression (LR), "Observed" plots the cumulative number of observed retweets by a given time. Notice SEISMIC quickly finds an accurate estimate of the tweet's final retweet count.

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. A cascade in the supercritical state is going through an "explosion" period and its final size cannot be predicted accurately at the current time. On the contrary, a cascade is tractable if it is in subcritical state. In this case, we are able to predict its ultimate popularity accurately by modeling the future cascading behavior by a Galton-Watson tree.

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 computation:** Making a prediction using SEISMIC only requires computational time linear in the number of observed reshares. Since predictions for individual posts can be made independently, our algorithm can also be easily 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 SEISMIC on one month of complete Twitter data, where users post tweets which others can then reshare by retweeting them. We demonstrate that SEISMIC is able to predict the final retweet count of a given tweet with 30% better accuracy than the state-of-the-art approaches (*e.g.*, [12]). For reasonably popular 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 dynamic list of 500 tweets over time, we are able to identify 78 of the 100 most reshared tweets and 281 of the 500 most reshared tweets in just 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 baselines as well as state-of-the-art approaches in Section 5. Last, in Section 6, we conclude and discuss future research directions.

2. RELATED WORK

The study of information cascades is a rich and active field [27]. Recent models for predicting size of information cascades are generally 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 [35], regression trees [3], content-based models [24], and passive-aggressive algorithms [26]. There are several issues with such approaches: laborious feature engineering and extensive training are crucial for their success, and the performance is highly sensitive to the quality of the features [4, 30]. Such approaches also have limited applicability because they cannot be used in real-time online settings—given the massive amount of posts being produced every second, it is practically impossible to extract all the necessary features for every post and then apply complicated prediction rules. In contrast, SEISMIC requires no feature engineering and results in an efficiently computable formula that allows it to predict the final popularity of millions of posts as they are spreading through the network.

The second type of approach is based on point processes, which directly models 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, 13, 14, 15, 18, 33, 36]. These methods have been successfully applied to study the spread of memes on the web [10, 14, 32, 33] as well as hashtags on

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: $\lvert\{i > 0; t_i \leq t\}\rvert$
R_∞	Final popularity (final number of reshares): $\lvert\{i > 0\}\rvert$
N_t	Cumulative degree of resharers by time t: $\sum_{i:t_i \leq t} n_i$
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.

Twitter [36]. In contrast, our goal is not to infer the network but to predict the ultimate size of a cascade in an observed network.

A major distinction between our model and existing methods based on Hawkes processes (*e.g.*, [22, 23, 33, 34, 36]) 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 [34] rely on computationally expensive Bayesian inference, while our method has linear time complexity. Another recently proposed related work is [12], which also takes the point process approach and directly aims 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 [12] in Section 5 and demonstrate a 30% improvement.

3. MODELING INFORMATION CASCADES

In this section, we describe SEISMIC and discuss how it can be used for:

1. Estimating the 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 an information cascade, which is measured by the ultimate number of reshares received by the post that started the cascade.

Important quantities in our model are the total number of reshares R_t of a given post up to time t and the cascade speed of spreading λ_t. In our model, λ_t is determined by the post infectiousness p_t and human reaction time. Our goal is to predict R_∞, the final number of reshares.

Another important quantity in our model is the *memory kernel* $\phi(s)$, which quantifies the delay between a post arriving to a user's feed and the user resharing it. Intuitively, infectiousness defines the probability that a given user will reshare a given post, and the memory kernel models user's reaction time. By combining the two we can then accurately model the speed at which the post will spread through the network. Table 1 summarizes the notation.

3.1 Human reaction time

In order to predict the cascade size, we need to know how long it takes for a person to reshare a post. Knowing the delay allows us to accurately model the speed of a cascade spreading through the network. We consider that the time s between the arrival of a post in a users' timeline and a reshare of the post by the user is distributed with density $\phi(s)$. The probability density $\phi(s)$ is also called a memory kernel because it measures a physical/social system's memory of stimuli [7].

The distribution of human response time $\phi(s)$ 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, 34]. However, due to the rapid nature of information sharing on Twitter, it is also natural to expect many instant 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 and then followed by a power-law decay. Different social networks may have different distributions of human reaction times. However, $\phi(s)$ only needs to be estimated once per network and thus we can safely assume it is given. We describe a detailed estimation procedure of $\phi(s)$ in Section 5.2.

3.2 Post infectiousness

The second component of our model is the 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 the post w is to be reshared at time t. Infectiousness of a post may depend on a combination of factors, including but not limited to the quality of the post's content, the social network structure, the current local time, and the geographical location. Instead of assuming a parametric form of p_t, we model it flexibly in a nonparametric way which implicitly accounts for all these factors.

Most existing methods studying self-exciting point processes assume p_t to be fixed over time. Consequently, an important concept is the *criticality* of the process R_t. In a self-exciting point process 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, no supercritical cascades can exist if p_t is assumed to be a constant. This is inadequate to model highly contagious tweets and our assumption of non-constant infectiousness solve this problem as well. Furthermore, as the post gets older the information becomes outdated and its spreading power (infectiousness) may decrease. This effect may also be observed as the post spreads further away from the original poster [34]. Alternatively, resharing by a highly influential user may increase the post's infectiousness. Thus, rather than assuming a common evolutionary pattern of p_t for all the tweets, we only assume it varies smoothly over time and use non-parametric methods to estimate p_t for each tweet.

3.3 The SEISMIC model

We combine human reaction times and post infectiousness to derive SEISMIC. In order to link p_t to the post resharing process R_t, we model R_t as a *doubly stochastic self-exciting point process*. This is an extension to the standard self-exciting point process (also called the Hawkes process [16]) which was initially used to model earthquakes [25].

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 [16]:

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

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

Note that the above point process is called *self-exciting* because each previous observation i such that $t_i \leq t$ contributes to the intensity λ_t, or equivalently, each observation increases the intensity in the future. It is further *doubly stochastic* (or a *Cox process*) because the infectiousness p_t is itself a stochastic process.

Additionally, we assume node degrees $\{n_i\}$ are independent and identically distributed with mean n^*. Mean degree n^* is related to the critical threshold p^* which is 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

In this section we describe how to perform statistical inference for the self-exciting model of information cascades introduced in the previous section. Specifically, we discuss how SEISMIC estimates the infectiousness parameter p_t and then predicts the ultimate size of the cascade R_∞.

Throughout this section, we make a technical assumption that the followers of all the resharers are disjoint, so we can use a tree structure to describe the information diffusion (Figure 2). The conclusions made in this section remain valid even if resharers are not disjoint. In this case, we can replace the node degree n_i with the total number of newly exposed neighbors of node i (the followers of i-th resharer who do not follow the first $i-1$ resharers).

4.1 Estimating post infectiousness

We first define the *sample-function density*, which plays a central role in estimating self-exciting point processes [29]. Let's denote $\mathcal{F}_t = \sigma\left(\{(n_i, t_i)\}_{i=0}^{R_t}\right)$ as the σ-algebra generated by all the information available by time t: the times t_i of all the reshares up to time t and the number of followers (*i.e.*, node degree) n_i of the i-th user to reshare. 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 where 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 [29, 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:

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

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 [29].

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

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

Thus, by assuming the infectiousness p_t to be a constant over time, one would essentially assume that most posts have the same infectiousness $1/n_*$. However, such assumption is unrealistic as it cannot explain the bursty and volatile dynamics information cascades (*e.g.*, Figure 1).

This undesirable consequence of assuming constant p_t is another motivation for allowing p_t to vary over time. To estimate p_t 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 kernels $K_t(s)$, $s > 0$, indexed by time t. We use these kernels to weight the reshares and the weighted estimate of p_t is given by

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

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 potentially 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 posts and gradually down-weights older posts. This keeps our estimator $\hat{p}(t)$ closer to the ever-changing true p_t. And last, as $K_t(s)$ 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 the one we use for Twitter in our experiments, see Section 5.

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 far the post is going to spread across the network.

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 the 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 that counts the reshare number of a post, as illustrated in Figure 2. Predictor for R_∞ used by SEISMIC can be stated as follows:

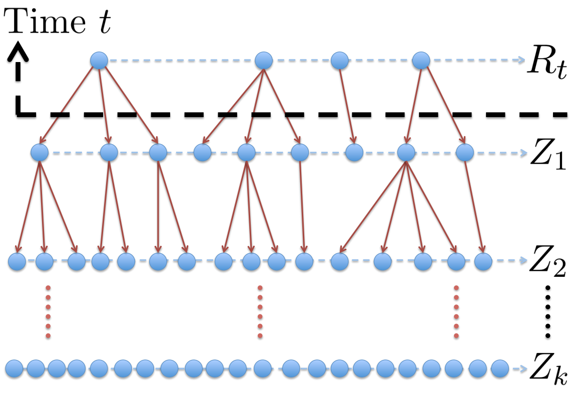

Figure 2: An illustration of the information diffusion tree. We observe the cascade up to time t (denoted by a dashed line) and the question is how the cascade tree is going to grow in the future. We define variables Z_k which denote the number of reshares caused by the k^{th} generation descendants. Using variables Z_k the final reshare count R_∞ can then be simply computer as $R_t + \sum\limits_{k=1}^{\infty} Z_k$.

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*

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

PROOF. First, we consider the case where $p < 1/n_*$. We define a sequence of random variables $\{Z_1, Z_2, Z_3, \ldots\}$ that models the future information diffusion tree, as illustrated in Figure 2. In this tree, Z_k denotes the number of reshares made by the k^{th} generation descendants (counting from generation R_t onward). Thus, the 1^{st} generation descendants Z_1 refers to the number of new reshares generated by the posts created before time t, the 2^{nd} generation descendants Z_2 refers to the reshares of the posts of the 1^{st} descendants, and so on. Notice that the summation over the Z_k's gives the post's final reshare count $R_\infty = R_t + \sum_{k=1}^{\infty} Z_k$. In the following we use descendants Z_k only for deriving Eq. (7) and emphasize that our final estimator does not require explicit network structure 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 the post gets. Using a standard branching process result, we have Z_i/μ^i is a martingale. Therefore, $\forall k > 1$, $\mathbb{E}\left[Z_{k+1}|Z_k\right] = \mu\, Z_k$, and,

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

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 $\mathbb{E}[Z_{k+1}] \geq \mathbb{E}[Z_k] \geq \cdots \geq \mathbb{E}[Z_1]$. Therefore the total future reshares $\sum_{k=1}^{\infty} Z_k$ has infinite expectation and the final reshare count cannot be reliably predicted. □

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 $i = 0, \ldots, R_t$.
**Algorithm:**
$N_t = 0$, $N_t^e = 0$
**for** $i = 0, \ldots, R_t$ **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:** $\hat{R}_\infty(t)$
```

Note that Prop. 4.1 assumes that the post infectiousness remains constant in the future ($p_s = p_t$ for $s \geq t$), which could be unrealistic for some information cascades. We correct this by changing the prediction formula in Eq. (7) by adding two scaling constants α_t, γ_t that adjust the final prediction:

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

We introduce these correction factors based on the following intuition. We expect α_t to decrease over time t so it scales down the estimated infectiousness in the future, which accounts for the post getting stale and outdated. Similarly, γ_t accounts for the overlap in the neighborhoods of reposters' followers. Over time as the post spreads farther in the network, we expect γ_t to increase as more nodes get exposed multiple times, which means the arrival rate of new nodes (previously unexposed nodes) decreases over time.

We use the same values of α_t and γ_t for all posts but allow them to vary 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 a training data set. As described in Section 5.2, we find α_t is more important than γ_t in practice.

4.3 The SEISMIC algorithm

Last, we put together all the components described so far and synthesize them in the SEISMIC algorithm. The 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. These 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 integration $\int_{t_i}^{t} K_t(t-s)\phi(s-t_i)ds$ and $\int_{t_i}^{t} \phi(s - t_i)ds$. However, the overall 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 $s_0 > 0$):

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

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 $i = 0, \dots, R_t$.
**Algorithm:**
$\tilde{R}_t = 0$, $\tilde{N}_t^e = 0$
**for** $i = 0, \dots, R_t$ **do**
    $\tilde{R}_t$ += $K_t(t - t_i)$
**end for**
**for** $i = 0, \dots, R_t$ **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:** $p_t$
```

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 piece-wise polynomials (polynomial with possibly non-integer exponents), which greatly decreases computational cost of SEISMIC.

5. EXPERIMENTS

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

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 dataset includes tweet id, posting time, retweet time, and the number of followers of the poster/retweeter. Note, the data set lacks Twitter network information. The only piece of network information available to us is the number of followers of a node.

We focus on a subset of reasonably popular tweets with at least 50 retweets, so that our model enables the prediction as soon as sufficient number of retweets occur. Note, that if multiple Twitter users independently post the same tweet, which then gets retweeted, each original posting creates its own independent cascade. All in all there are 166,076 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

First we describe how to fit the memory kernel $\phi(s)$ (Section 3.1). We carefully choose 15 tweets in the training set and use the distribution of all their retweet times as our $\phi(s)$ (Figure 4). The histograms of the 15 sequences of retweet times all display a clear shape of subcritical decay. Moreover, all the original posters have an overwhelming number of followers. Therefore, most of the retweets, if not all, should come from the immediate followers of the original poster. Consequently, the distribution of human reaction time can be well approximated by that of the retweet times of these 15 tweets. The estimation of $\phi(s)$ can be further improved if the network structure is available.

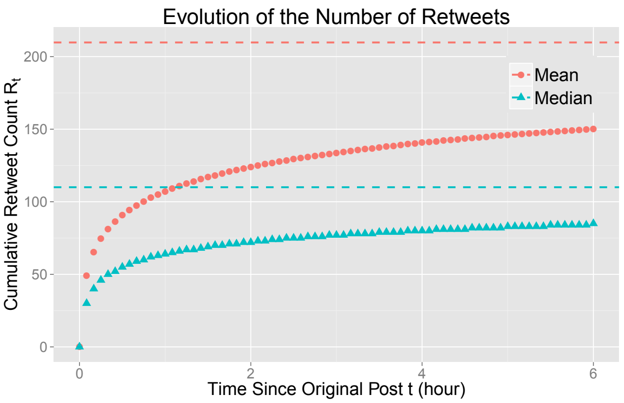

Figure 3: Convergence of the mean and media cumulative retweet count R_t as a function of time.The horizontal lines correspond to mean and median final retweet count $R_{14\text{ days}}$. On average, a tweet receives 75% of its retweets in the first 6 hours.

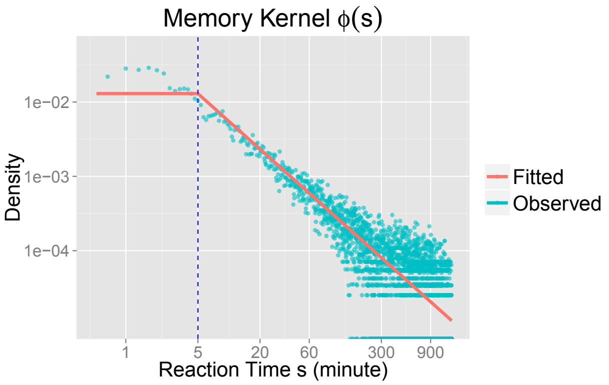

Figure 4: Reaction time distribution and the estimated memory kernel $\phi(s)$. The reaction time is plotted on logarithmic axes, hence the linear trend suggests a power law decay.

The observed reaction time distribution (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 estimate power law decay parameter $\theta = 0.242$ with the complimentary cumulative distribution function (ccdf), and chose $c = 6.27 \times 10^{-4}$ to make $\int_0^\infty \phi(s)ds = 1$. The memory kernel is a network wide parameter and 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

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: Values α_t used in Algorithm 1.

slow and consistent decay to account for the fact that information is getting increasingly stale and outdated over time.

With all the estimated parameters in SEISMIC, we are ready to apply it (Algorithms 1 and 2) to the Twitter dataset. For a given tweet w and every 5 minute interval t, we output our estimate $\hat{R}_\infty(t, w)$ of the tweet's final retweet count $R_\infty(w)$.

5.3 Baselines for comparison

We consider four different prediction methods for comparison. The first two are regression based and the next two are point process based.

- **Linear regression (LR) [31]:** 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 [34]. Notice that all the tweets receive the same multiplicative constant for a given time.

- **Linear regression with degree (LR-D) [31]:** 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 $\log R_t$ to have a slope not equal to 1 and uses additional features.

- **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 the retweet times into $b = 5$ minute intervals. Note that when $\gamma > -1$, the integral $\int_{t_{\text{peak}}+b}^{\infty} \lambda_t dt$ is infinite. In such cases, we move t_{peak} forward to the second maximum bin.

- **Reinforced Poisson Model (RPM) [12]:** This recently published state-of-the-art approach models the reshare rate 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 out of range.

5.4 Evaluation metrics

For a particular tweet, suppose that the prediction for R_∞ at time t is denoted by $\hat{R}_\infty(t)$. We use the following evaluation metrics in our experiment:

- **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 the 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 [19], 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 of values of $\hat{R}_\infty(t)$ and R_∞. A high value of rank correlation means the predicted 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 methods by quantifying how well the predicted top-k list covers the ground-truth top-k list. We give additional details in Section 5.5.3.

5.5 Experimental results

In this section we evaluate the performance of SEISMIC and the four competitors 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 empirically validate SEISMIC. In Proposition 4.1, we obtain a formula for the expected number of final retweets in terms of the infectiousness parameter p_t. Our goal here is to show that Proposition 4.1 provides 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 diagonal 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 the 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, in practice one often wants to shrink the prediction in order to stabilize the estimator and achieve better overall performance. Therefore, we use the calibrated prediction formula Eq. (8) for the rest of the experiments.

5.5.2 Predicting final retweet count

We run our SEISMIC method for each tweet and compute the Absolute Percentage Error (APE) 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 =10 min), 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 the tweets. 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 improvement over the baselines and the state-of-the-art as shown in Figures 7 and 8. The left panels of Figures 7 and 8 show the median APE of different methods over time as more and more of the retweet cascade gets revealed. The LR and LR-D baselines have very similar performances, 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 approach RPM is worse than LR and LR-D in the early period but becomes better after about 2 hours. All in all, in terms of median APE score SEISMIC is about

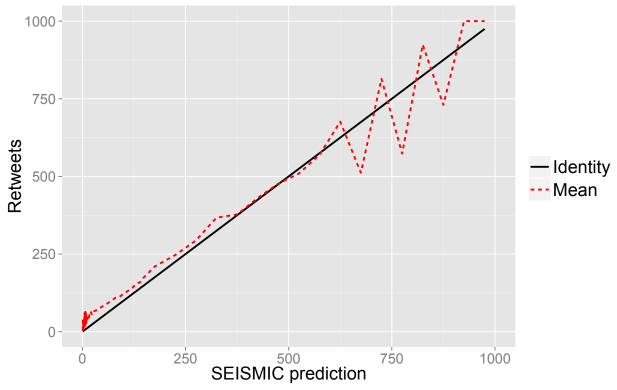

Figure 5: Predicted final retweet counts nicely follow the ground-truth retweet counts, which suggests SEISMIC provides an unbiased estimate of the final retweet count. The dashed red curve is obtained by binning the tweets according to the prediction and then computing the average number of retweets in each bin.

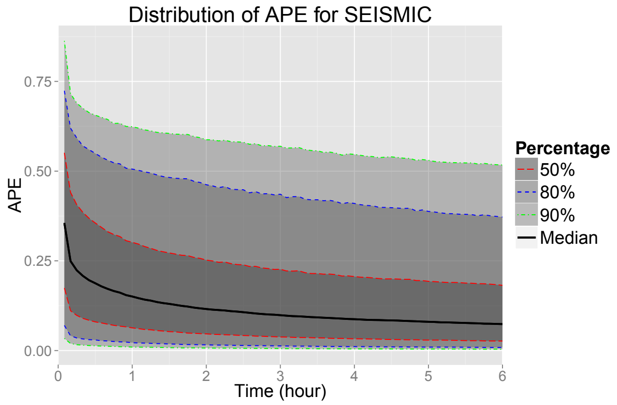

Figure 6: Absolute Percentage Error (APE) of SEISMIC on the test set. We plot the median and the middle 50th, 80th, 90th percentiles of the distribution of APE across the tweets.

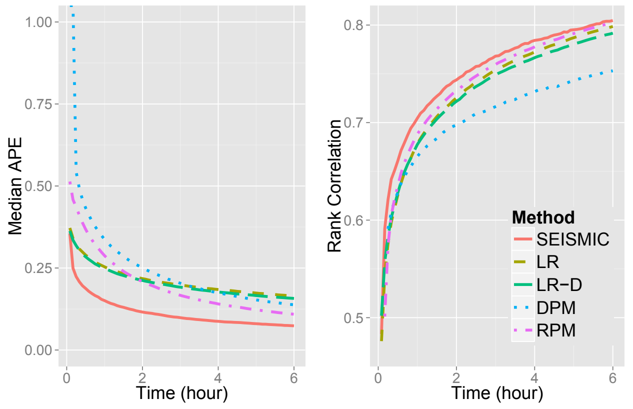

Figure 7: Median Absolute Percentage Error (APE) and Kendall's Rank Correlation of SEISMIC and the baselines as a function of time. SEISMIC consistently gives best performance.

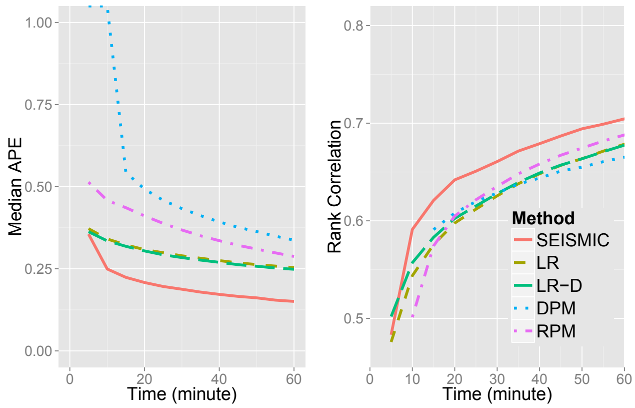

Figure 8: Zoom-in of Figure 7: Median APE and Rank Correlation for the first 60 minutes after the tweet was posted. SEISMIC performs especially well compared to the baselines early in the tweet's lifetime.

30% more accurate than all the competitors across the entire twee lifetime.

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 ground-truth ranking of tweets. Again SEISMIC is giving much more accurate rankings than other methods.

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 to as early as possible identify "breakout" tweets, which have the highest final retweet count. We quantify the performances of different models in detecting breakout tweets by using models' predictions of tweets' final retweet counts.

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

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 $\hat{L}_m(t)$.

Fig. 9 shows the performance of SEISMIC in 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 fifth most retweeted tweet in this plot is actually the tweet we showed earlier in Figure 1. We observe that SEISMIC detects this tweet 30 minutes after it has been posted, while LR and LR-D both take more than an hour. DPM fails to detect this breakout for the first 6 hours (plots not show for brevity).

To compare SEISMIC with other 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, SEISMIC covers 65% of the shortlist, while LR-D and LR both cover only 50%. In

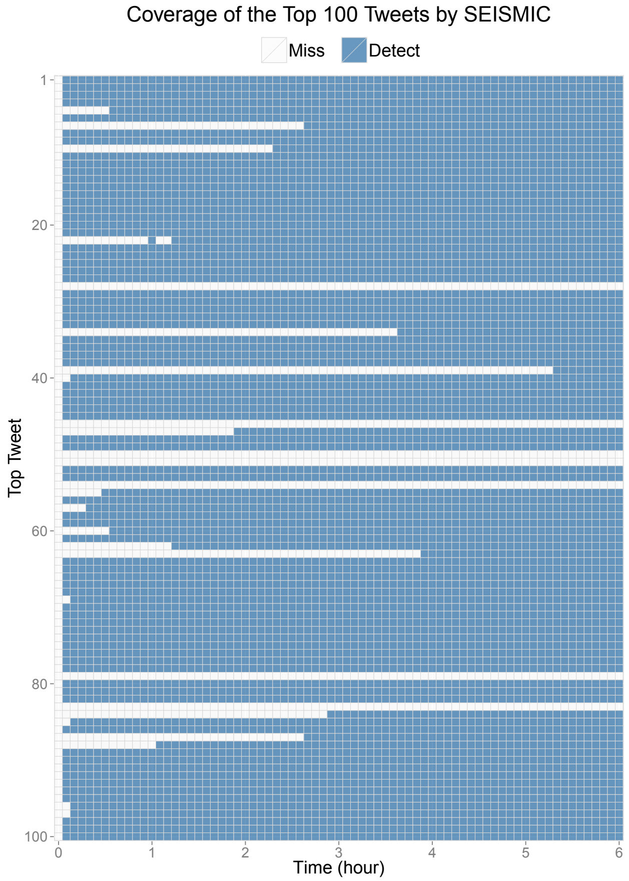

Figure 9: Coverage of top 100 most retweeted tweets. Each row represents a tweet. White blocks indicate that a given tweet was not covered by SEISMIC's predicted list of top-500 tweets at time t, and blue indicates successful coverage.

general, the dynamic Poisson model fails 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 counts.

5.6 Discussion of model robustness

SEISMIC demonstrates better robustness than the other two point process based methods — DPM and RPM. While SEISMIC is not able to make a prediction for tweets that are in the supercritical state, DPM and RPM are unable to make predictions when the decay parameter is outside the feasible 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 70min, DPM fails to make a prediction for the first 6 hours and RPM is only able to make a prediction from 30 to 80 minute.

All in all, we find that tweets are in the supercritical regime for only a very short time and SEISMIC is able to make predictions for most of the tweets in most of the time. We find that on average, SEISMIC is not able to make a prediction for 1.80% of the tweets after observing them for 15 minutes. In other words, after 15 minutes, 1.80% of the tweets are still in the supercritical regime (over all the tweets with at least 50 retweets). This number drops to 1.29% (0.67%) after 1 hour (6 hours). As a point of comparison we also note that other methods are not able to make predictions for a much larger fraction of tweets: DPM fails to make a prediction for 6.77%, 5.79% and 1.45%, and RPM fails for 3.45%, 5.69% and 15.43% of the tweets after 15min, 1h, and 6h.

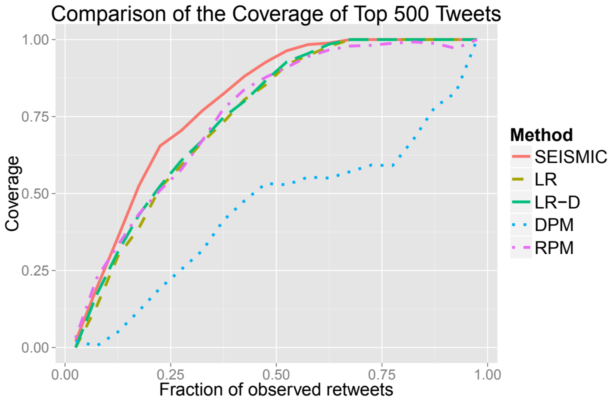

Figure 10: Coverage of top 500 tweets (L^*_{500}) by various methods. SEISMIC exhibits clear improvement over all methods after about 10% of retweets are observed. All methods except DPM achieve perfect coverage after 65% of retweets are observed.

Our SEISMIC method is also significantly faster than the RPM model [28], 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.

6. CONCLUSION AND FUTURE WORK

In this paper we propose SEISMIC, a flexible framework for modeling information cascades and predicting the final size of an information cascade. Our contributions are as follows:

- We model the information cascades as self-exciting point processes on Galton-Watson trees. Our 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. This 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 network structure.

There are many interesting venues for future work and our proposed model can be extended in many different directions. For example, if the 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 develop a content-based prior of p_t for each post. If temporal features such as the users' 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.

SEISMIC is 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, paving ways towards better management of shared content.

Data and Software

The SEISMIC software and the dataset we use in Section 5 can be found in `http://snap.stanford.edu/seismic/`. The R package of our algorithm is also available on `http://cran.r-project.org/web/packages/seismic`.

Acknowledgements

The authors would like to thank David O. Siegmund for his constructive suggestions and Austin Benson, Bhaswar B. Bhattacharya, Joshua Loftus for their helpful comments. This research has been supported in part by NSF IIS-1016909, CNS-1010921, IIS-1149837, IIS-1159679, ARO MURI, DARPA SMISC, DARPA SIMPLEX, Stanford Data Science Initiative, Boeing, Facebook, Volkswagen, and Yahoo.

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. In *WWW '09*, 2009.

[3] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *WSDM '11*, 2011.

[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? In *WWW '14*, 2014.

[7] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 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. In *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. In *NIPS '12*, 2012.

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

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

[13] M. Gomez-Rodriguez, J. Leskovec, D. Balduzzi, and B. Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2:26–65, 4 2014.

[14] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML '13*, 2013.

[15] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Structure and Dynamics of Information Pathways in On-line Media. In *WSDM '13*, 2013.

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

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

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

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

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

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

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

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

[30] 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.

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

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

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

[34] 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.

[35] 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.

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