# Detecting Large Reshare Cascades in Social Networks

Karthik Subbian[†], B. Aditya Prakash[*], Lada Adamic[†]
[†]Facebook Inc.
[*]Department of Computer Science, Virginia Tech.
Email: {ksubbian, ladamic}@fb.com; badityap@cs.vt.edu

## ABSTRACT

Detecting large reshare cascades is an important problem in online social networks. There are a variety of attempts to model this problem, from using time series analysis methods to stochastic processes. Most of these approaches heavily depend on the underlying network features and use network information to detect the virality of cascades. In most cases, however, getting such detailed network information can be hard or even impossible.

In contrast, in this paper, we propose SansNet, a network agnostic approach instead. Our method can be used to answer two important questions: (1) Will a cascade go viral? and (2) How early can we predict it? We use techniques from survival analysis to build a supervised classifier in the space of survival probabilities and show that the optimal decision boundary is a survival function. A notable feature of our approach is that it does not use *any* network-based features for the prediction tasks, making it very cheap to implement. Finally, we evaluate our approach on several real-life data sets, including popular social networks like Facebook and Twitter, on metrics like recall, F-measure and breakout coverage. We find that network agnostic SansNet classifier outperforms several non-trivial competitors and baselines which utilize network information.

## 1. INTRODUCTION

Every day millions of users engage and express their feedback on social networking platforms like *Twitter.com* and *Facebook.com*, using a *like*, *comment* or *share*. While likes and comments provide feedback, it is resharing that has the potential to spread information to millions of users in a matter of few hours or days. Such a spread of information through resharing is called a cascade [7]. Most of the cascades do not spread far and beyond but are restricted to only a small group of people and hence remain very small in size [4]. However, very few of them (far less than 1%) become substantially big and are referred to as *viral* cascades. Knowing whether something is going viral can be valuable and there are several recent related work that deal with controlling or accelerating such cascades in various scenarios [23, 31].

*WWW 2017,* April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052718

Understanding viral cascades has several challenges. Foremost, there is lack of knowledge of complete network structure, through which the information propagates [10]. This may be due to a variety of reasons. Population networks and malware infection networks are hard to gather and expensive to construct [2]. For data like blog cascades, most networks are only inferred networks [9]. Edges in online social networks are also typically unavailable for prediction. As a result, either a network is unavailable, or difficult to obtain, or noisy. In this paper, hence, we address the problem of understanding cascades without using the network structure and instead modeling it purely as a time series. Time series modeling is particularly practical because, e.g. the number of reshares on Twitter or Facebook for a piece of public meme can be readily retrieved using search API's.

Further, for many practical purposes, knowing whether a cascade would grow big or not is *more useful* than actually measuring the exact size of the cascade (which is much harder indeed). Such a prediction is useful if we can predict virality *much earlier* in its life than much later. There are two important characteristics of the viral cascade prediction problem that must be modeled carefully. First, viral cascading phenomena are generally rare and this rarity makes the class distribution particularly skewed. Some approaches consider under- or over-sampling the data to account for the skewness, especially when the problem is treated in a binary classification set up. Second, the time at which the event occurs is extremely important and detecting the virality earlier is much more important than predicting the accurate size much later.

Traditional time-series methods [3] are ill-suited to modeling the above two characteristics of the problem, primarily because they ignore diffusion dynamics. For example, autoregressive and exponential smoothing models do not work for our scenario as the reshare time-series is bursty and does not exhibit seasonality or trend correlations with historical data. Some also try to incorporate the bursty characteristics of the time series [21, 34], but they suffer from assuming specific models for node influence or availability of network.

In contrast, survival analysis is extremely suited to predict virality for several reasons. There is no need to under- or over-sample the data and the actual class distribution plays an important role in modeling the survival probabilities. Moreover, survival models can accommodate incomplete (or censored) data during training. So there is no need to eliminate the “just started” or very small cascades from the training data (as the first few hours play an important role in the diffusion dynamics of the cascade), as other methods do [37]. Finally, survival probabilities are computed using hazard rates which are robust estimators of instantaneous rates of change. Hence, we do not need to assume a particular model, and a sudden rate of change in a few cascades will not affect the survival probabilities.

We propose a *classification method* SansNet, in this paper, based on the survival probabilities to estimate the non-linear decision boundary to separate the viral cascades from the non-viral ones. The estimated decision boundary turns out to be a survival function and it can model the cascades that may never go viral by incorporating them as ‘right-censored data’ (i.e. data points for which technically the time of death is unknown). Our empirical tests show that SansNet is very fast, and outperforms several baselines on multiple real life social networks, including those which utilize network information. Again, as discussed before, we predict *virality* instead of predicting the actual size as many other methods. Hence we believe this gives a novel complementary viewpoint to current approaches.

This paper is organized as follows: We present the related work in Section 2. In Section 3, we present the required preliminaries for survival analysis, followed by the problem definition and our approach to learn the survival separation boundary. In Section 4, we present the empirical results and we conclude in Section 5.

## 2. RELATED WORK

We focus on related work from the following main areas: cascade analysis, survival analysis and time-series prediction, epidemiology, and optimization problems. In general, there has been a lot of interest in studying dynamic processes on large graphs like (a) blogs and propagations [16, 19], (b) information cascades [8, 27]; (c) marketing and product penetration [26, 28] and (d) malware prediction [22].

**Cascade Analysis:** Methods for predicting size of information cascades are generally characterized by two types of approaches, feature based methods and model-based methods. Feature based methods [32, 4] compute an exhaustive list of potentially relevant features and use them in a classification setting. There are several drawbacks with these approaches, including laborious feature engineering, extensive training, scalability issues in terms of computing these features at scale and in an online manner (c.f. the Wiener index [32]). In contrast we only use reshare counts per unit time, which are very cheap to get with today’s online data aggregation services. There are model based methods [6], where the model predicts whether the cascade will go above a certain size threshold. In a recent work [6], Cui et al. proposes a logistic model considering all nodes as features. The model in this case measures the relative importance of each node given the nodes that have propagated before them. One of the drawbacks of this approach is to maintain the status of cascade across all nodes in a network and it can be particularly difficult when the number of nodes is in billions. We overcome such difficulty by summarizing the cascade growth at certain frequency.

The second broad thread is by using models which generatively explain the information cascade process. Such methods typically take inspiration from epidemiological models [21], and have been used to model the spread of memes or hashtags [35, 34], Youtube views [5] or even keyword volume [21]. Many recent methods rely on using stochastic point processes like the Hawkes self-exciting processes [37, 25] or personalized behavioral dynamics [36]. However many of these approaches [5, 37] either silently assume infinite available nodes, or are not able to make a prediction if the cascade is in the ‘super-critical’ state.

In both these broad approaches however, access to the underlying network is assumed, which in reality as discussed before, is not readily available or can be noisy. Further, many of them have been designed to predict the size of the cascade, rather than the more practically relevant question of if and when it will go viral. Hence, we instead develop a network-agnostic approach which avoids these issues.

**Survival Analysis and Time-series Prediction:** Survival analysis is often used to model the time to event data, such as death, infection, or diagnosis of a type of cancer [15]. It is therefore fundamental to many medical studies. In the recent past, survival theory has been used to infer the unobserved network either using an additive or multiplicative risk model [10]. Here the hazard rate of each node is an additive (or multiplicative) function of infection times of previously infected nodes. Based on the hazard rates of individual nodes an estimate of an edge being incoming or outgoing is determined. An important take way from this work is that either observing the complete network or inferring them is a hard problem. In another recent work [33], survival analysis is used to predict the number of actors that would be mentioned in a Tweet and the length of the influence chain from that actor. This work does not deal with predicting virality of the cascade in terms of its size. Survival models are also used in user return time prediction based on their historical usage patterns [13]. However, none of these works use a supervised learning approach for classification (like max-margin classifier) using survival probabilities to address their prediction problem.

Auto-regressive models and exponential smoothing models are the first-hand approaches for modeling a time-series [3, 20]. However, the time series of reshare cascades are bursty in nature and the correlation between historical and future data becomes difficult to be captured in these regression-based approaches.

**Epidemiology:** The classical texts on epidemic models and analysis are [1, 11]. Most work in epidemiology is focused on *homogeneous models*. Much work has gone into finding epidemic thresholds for networks (minimum virulence of a virus which results in an epidemic) for a variety of virus propagation models [24].

**Optimization Problems:** There exist several diffusion based optimization problems, including the influence maximization problem, formulated by Kempe et. al. [14] as a combinatorial optimization problem. Other such problems where we wish to select a subset of ‘*important*’ vertices on graphs, include ‘outbreak detection’ [18], ‘finding most-likely culprits of epidemics’ [17], and immunization [23].

## 3. MODELING SURVIVAL SEPARATION

We consider a cascade to have gone viral if it crosses a specific relative size threshold (how to set it will be discussed later). Let us first understand how survival analysis applies to our problem. Consider a person accumulating infection over time, and when the infection level cross the immunity threshold, the person eventually dies due to the infection. This is analogous to the viral cascade situation in our problem. Consider each cascade to be a person and the size of the cascade (or number of shares) is the infection. When the size crosses a specific threshold, the cascade has gone viral.

It is easy to see that most of the cascades never cross the threshold. Hence, these cascades where the event never occurred (equivalent to the person survived in our example) add additional information to the model about the rarity and timing of the actual event. These observations are included in the model as right-censored observations and hence no over- or under-sampling is required in our approach compared to other classification approaches dealing with rare-class classification.

## 3.1 Problem Formulation

**What is Viral?** The problem of predicting the survival probabilities of each cascade depends on how well we define the virality. The most common and useful definition is based on its size [4]. As discussed earlier, for most practical purposes, predicting the exact size is far less important and also harder than knowing whether a cascade would go above a certain size threshold. The threshold can be set either as a relative measure or absolute measure. For instance, setting a fixed threshold of 10,000 reshares is an absolute threshold, while setting it to 90-th percentile of the data is a relative measure. Absolute measures are invariant to the dynamics of the new observations. Relative measures are useful in cases when the audience size is unknown and it is relative to population observed in the latest data (especially if the user engagement with the social network is changing). Our method is agnostic to relative or absolute thresholds and we leave it to the choice of the exact application.

**Definition** Let us now formally define the problem. Let $\tau$ be the threshold on the size of the cascade, above which a cascade is considered to be viral. Given hourly reshares of $m$ posts (i.e. $m$ time series) from the beginning of their time until the end of the observation period, our goal is to predict whether a partially observed cascade at $j$-th hour will become viral? Formally, let $x_i$ be the reshare-count time series of the $i$-th cascade and $x_i(j)$ be the number of reshares for cascade $i$ at time $j$. The current size of the cascade at time $t$ is denoted by $v_i(t) = \sum_{j=1}^{t} x_i(j)$. Our problem is formally defined as follows.

Problem 1 (Viral Cascade Prediction). *Given the cascade virality threshold $\tau$, and $m$ time series $x_1, \ldots, x_m$ of arbitrary length $l_1, \ldots, l_m$ with same sampling rate, and a test time series $z$ of length $p$, s.t. $\sum_{j=1}^{p} z(j) < \tau$, predict whether $\sum_{j=1}^{p} z(j) + \sum_{k=p+1}^{q} \hat{z}(k) >= \tau$, for some $q > p$.*

Note that as $q >> p$ it is harder to make future predictions, particularly when $\tau - \sum_{j=1}^{p} z(j) >> 0$. In other words, with little or almost no information about the cascade, it is hard to guess whether it would go viral or not. On the other hand, when $q - p \approx 1$ and $\tau - \sum_{j=1}^{p} z(j) \approx 0$ it is much easier to make the prediction, as it is most likely to be a viral cascade. Note that commonly used measures like absolute percentage error (or other equivalents) are ill-suited for such a prediction approach, as it does not score results based on the real distribution of easy versus hard cascade predictions.

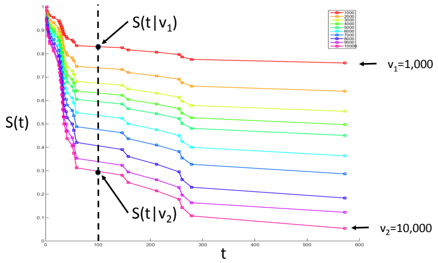


**Figure 1: Estimated survival function for various sizes.**

## 3.2 Estimation and Separation

Let us denote the time to event with a random variable $Y$, and the possible values include all non-negative numbers. Any specific value of this random variable is denoted by $t$. The survival function $S(t)$ gives the probability the cascade will *not* encounter an event until time $t$, i.e. $S(t) = P(Y > t)$. Survival functions are monotonically decreasing in nature. Again, note that the use of term *survive* in this paper implies that the event *did not* happen and in our case it is equivalent to the cascade that *did not go* viral.

**Estimation** We use the Cox-extended model [15] to estimate the survival function for $m$ different time series. The Cox-extended model is particularly useful for our problem as it allows for estimating the survival function, while controlling for time varying covariates (unlike a Cox PH model [15]). The survival estimator for this model is:

$$S(t|v(t)) = e^{-\hat{H}(t|v(t))} \tag{1}$$

The time varying covariate in our case is the size of the cascade ($v(t)$) at time $t$. A unique integration [15] can be used to estimate the cumulative 'hazard' estimator $\hat{H}$ in this case. For all practical purposes, the estimator is available in statistical softwares like $R$[1]. To learn more about survival analysis or estimation of Cox extended models we ask the readers to refer [15].

Intuitively, as the size of the cascade increases the survival probability drops at any given time $t$, i.e. $S(t|v_1) \leq S(t|v_2)$, where $v_1 \geq v_2$. As we observe more and more of the cascade over time, the survival function models the rate at which the survival probabilities drop as a function of time, conditioned on the current size of the cascade. Note that if we have a better predictor of future size of the cascade then it would nicely complement our approach, in that, it will be able to tell the survival probability of the future time point, given size at that time point.

**Role of size** The size covariate plays an important role in modeling the survival function. We trained a Cox-extended estimator on the Facebook dataset (explained later), and computed the survival probabilities over time for various size values (1K to 10K in steps of 1K). Each curve in Fig. 1 represents the survival probabilities for a given size. It is very evident from Fig. 1 that as the size of the cascade increases the estimated survival probability drops significantly at some time point $t$. The drop in the earlier time points are relatively smaller compared to the later time points. This is because the probability of a cascade going viral earlier is much less compared to a later time.

[1] `https://cran.r-project.org/web/packages/survival/survival.pdf`

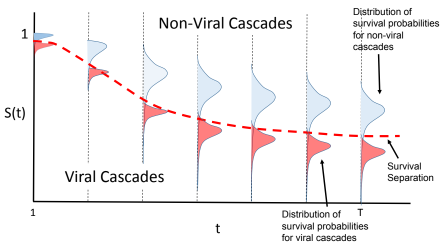

Non-Viral Cascades
Distribution of survival probabilities for non-viral cascades
1
S(t)
Viral Cascades
Survival Separation
1
t
Distribution of survival probabilities for viral cascades
T

**Figure 2: An illustration of survival separation function to separate the viral cascade survival probability distribution from the non-viral ones.**

**Separation** How do we use these dynamic survival probabilities for the purpose of classification (viral or not)? Let $S_i(t)$ denote the estimated survival probability of cascade $i$ at time $t$. Consider the distribution of survival probabilities $S_i(t)$ at any time point $t$ across all the cascades $i$. Then the tail of the survival distribution characterizes the viral cascades (with *low* survival probabilities). This is particularly useful in an *unsupervised classification* setting. However, we know the class labels (viral or not) for each of the observed cascades. We use these labels to separate the survival distribution of viral and non-viral cascades at each time point, in a *supervised manner*. This is illustrated using a graphic in Fig. 2. The plot shown is an illustration—however, the actual distribution of survival probabilities is non-parametric and does not make any model assumptions.

Our proposal is to find the optimal survival function separating the two classes using the following optimization:

$$\begin{aligned} &\underset{\hat{S}=s_1,\ldots,s_T}{\text{minimize}} && \sum_{t=1}^{T}\sum_{i=1}^{m} c_i \cdot y_i \cdot \text{sgn}(S_i(t) - \hat{s}_t) \\ &\text{subject to} && \hat{s}_{t+1} - \hat{s}_t \leq 0, \forall t = 1, \ldots, T-1 \end{aligned} \quad (2)$$

In this formulation $y_i$ is the class label of the $i$th cascade being viral $(+1)$ or not viral $(-1)$, i.e. $y_i \in \{-1, +1\}$ and $c_i$ is the cost of mis-classification. We account for the skewness in the class distribution by setting $c_v = m_n/m$ and $c_n = m_v/m$ as the mis-classification cost for viral and non-viral class data points respectively. Here $m_n$ and $m_v$ are the number of training data points in non-viral and viral classes respectively.

The objective of Eq. (2) is to *maximize the separation between the two classes using the survival probabilities* or in other words minimize the mis-classification cost. The constraint ensures that the resulting separation boundary $S^*$ is a survival function and satisfies monotonicity property. This is important as it minimizes the training error and ensures that the resulting decision function is a survival function. In other words, if a cascade is non-viral at time $t$ and it does not increase in size further, then it should remain non-viral at $t' > t$. Note that in our analysis and formulation we have discrete time points (and not continuous), since the underlying time series is sampled at some rate. Also, the survival boundary $S^*$ is estimated until the end of observation period $T$, i.e. $S^* = s_1^*, s_2^*, \ldots, s_T^*$.

It is easy to see that, if we did not have the monotonicity constraint, then Eq. (2) can be broken down in to $T$ independent sub-problems. Each of this sub-problem is linear in nature and requires $O(m)$ time computation to evaluate the optimal value. However, let us say, we solve the constrained version using an unconstrained Lagrangian, the objective becomes combinatorial in nature. Consider the case where we find the optimal value for $s_T^*$ and start to back-solve for $T-1$. If we find that $s_{T-1}^* < s_T^*$, then $s_T^*$ has to be recomputed, since it violates monotonicity. In a brute force manner, there will be $O(m^T)$ computations required to find the optimal solution.

As a matter of fact, we know that each of the $m$ input survival functions corresponding to each input time series is monotonic [15]. Let us use this fact and show next (in Theorem 1) that the resulting decision boundary is also a survival function, then the constraint becomes trivial and the problem can be solved in linear time.

## 3.3 Analysis

Let $F_v^t(s)$ be the fraction of *viral* data points that fall below the decision boundary $s$ at time $t$, i.e. $F_v^t(s) = |\{i \mid (S_i(t) < s) \wedge (y_i = +1)\}|/m_v$. Similarly, for non-viral class, $F_n^t(s) = |\{i \mid (S_i(t) < s) \wedge (y_i = -1)\}|/m_n$. Clearly, both the functions are non-decreasing in $s$.

Lemma 1. *For any given survival probability* $0 \leq s \leq 1$, *$F_v^t(s)$ and $F_n^t(s)$ non-decreasing with $s$. That is,*

$$F_v^t(s) \geq F_v^t(s'), \forall s \geq s', \forall t$$
$$F_n^t(s) \geq F_n^t(s'), \forall s \geq s', \forall t$$

Using Lemma 1, we can see that both the fraction of viral and non-viral points increases at time $t$, for any $s \geq s'$. Moreover, we also observe from the data that this increase in the fraction of non-viral points, at a future time $(t' > t)$, is always higher than the increase in the fraction of viral points, with reference $s_t$. In other words, the survival probability of non-viral points do not decrease towards zero as rapidly as the viral ones. Formally, we specify this condition in Observation 1.

Observation 1. *Given an optimal separation point $s_t$ at time t, $F_v^{t+1}(s_{t+1}) - F_v^{t+1}(s_t) \leq F_n^{t+1}(s_{t+1}) - F_n^{t+1}(s_t), \forall s_t - s_{t+1} \leq 0$ and $\forall t \geq 1$.*

Theorem 1. *The optimal survival boundary solving Eq. (2) $S^* = s_1^*, s_2^*, \ldots, s_T^*$ is itself a survival function.*

Proof (Sketch). Let $m_{v+}$ $(m_{n+})$ and $m_{v-}$ $(m_{n-})$ be the correctly and incorrectly classified viral (non-viral) data points given the decision boundary $s_t^*$. Then the objective function in Eq. (2) can be written for some particular time $t$ and decision point $s_t^*$ as,

$$g_t(s_t^*) = \min\left[c_v m_{v-} - c_v m_{v+} + c_n m_{n-} - c_n m_{n+}\right].$$

Adding and subtracting $c_v m_{v+}$ and $c_n m_{n+}$, we get,

$$g_t(s_t^*) = \min\left[c_v m_v + c_n m_n - 2c_v m_{v+} - 2c_n m_{n+}\right].$$

Using $F_v^t(s_t^*) = m_{v+}/m_v$ and $F_n^t(s_t^*) = 1 - m_{n+}/m_n$,

$$g_t(s_t^*) = \min\left[K - 2c_v m_v F_v^t(s_t) + 2c_n m_n F_n^t(s_t)\right].$$

where $K$ is a non-negative constant independent of $t$ and $s$. Next, note that in our setting $(c_v = m_n/m$ and $c_v = m_v/m)$, the total mis-classification cost is balanced between the two classes i.e. $c_v m_v = c_n m_n = m_v m_n/m = \alpha$. So we get $g_t(s_t^*) = K - 2\alpha(F_v^t(s_t^*) - F_n^t(s_t^*))$.

Equivalently, the mis-classification cost at $t+1$ for the points $s_t^*$, and $s_{t+1}^*$ can be written as, $g_{t+1}(s_t^*) = K - 2\alpha(F_v^{t+1}(s_t^*) - F_n^{t+1}(s_t^*))$ and $g_{t+1}(s_{t+1}^*) = K - 2\alpha(F_v^{t+1}(s_{t+1}^*) - F_n^{t+1}(s_{t+1}^*))$ respectively. Subtracting these two costs at $s_t^*$

and $s^*_{t+1}$ we get,

$$\begin{aligned}
&g_{t+1}(s^*_t) - g_{t+1}(s^*_{t+1})\\
&= 2\alpha[(F_v^{t+1}(s^*_{t+1}) - F_n^{t+1}(s^*_{t+1})) - (F_v^{t+1}(s^*_t) - F_n^{t+1}(s^*_t))]\\
&= 2\alpha[(F_v^{t+1}(s^*_{t+1}) - F_v^{t+1}(s^*_t)) - (F_n^{t+1}(s^*_{t+1}) - F_n^{t+1}(s^*_t))]\\
&= 2\alpha\beta
\end{aligned}$$

Here $\alpha > 0$, but $\beta < 0$ from Observation 1, for all $s^*_{t+1} > s^*_t$. Hence for all $s^*_{t+1} > s^*_t$, we can write $g_{t+1}(s^*_t) - g_{t+1}(s^*_{t+1}) \leq 0$. Thus there does not exist an optimal survival decision boundary point $s^*_{t+1}$ at $t+1$, such that $s^*_{t+1} > s^*_t$ and $g_{t+1}(s^*_{t+1}) \leq g_{t+1}(s^*_t)$, as $s^*_t$ is more optimal at $t+1$ in such case. Hence $s^*_{t+1} \leq s^*_t, \forall t > 0$. □

### 3.4 Early Prediction

Early prediction of virality has immense practical value like that of predicting the virality of cascade itself. It can be particularly useful to counteract spam or malicious content before it becomes viral and infects thousands of users. Our approach is flexible enough to answer not only whether a cascade would go viral, but also *how early can we detect it.*

We have, so far, estimated the optimal survival boundary $S^*$ by minimizing the mis-classification error, or in other words, maximizing the margin between positive and negative classes. Given a partially observed time series $z$ until time $t$, we can compute the current size of the time series as $v_z = \sum_{j=1}^{t} z(j)$. We can measure how *early* each approach can predict by measuring the performance at different times before the cascade became viral (actual virality). Additionally, we can also calculate the *Early Prediction Advantage* (EPA). For the current size, compute the first time point $t^* > t$ at which the estimated survival probability for $z$ in the future is less than $s^*_t$, i.e.

$$t^* = \arg\min_{\hat{t}>t} \mathbb{1}(s^*_t > S(\hat{t}|v_z)) \qquad (3)$$

The difference between the actual time of virality and $t^*$ is the EPA for that cascade.

### 3.5 SansNet: Algorithm

In this section we describe the algorithm for our approach, SansNet (see Figure 3).

The key part of the algorithm is computing the risk intervals for each time series. This is essential for performing the Cox-extended estimation as the hazard function will change at the end of each risk interval. A risk time point is when a death (+1) or a right-censoring was observed in the training data. For example, if there are $p$, $q$ and $r$ risk time points, then following three risk intervals will be constructed $[0, p), [p, q), [q, r)$. Then for each of these intervals we need to compute the total reshare size for $x_i$ at the end of the interval. Then a table ($D$) containing the id of the cascade, the start time of the interval, end time of the interval, current size at the end of interval, and ground truth label can be used to estimate the survival model $M$ using the `coxph` routine [30] in R (or other statistical packages like SAS).

Once the survival model $M$ is computed, for each of time points $t$ we compute the survival probabilities for all $m$ time series based on its current size at $t$. We then use Eq. (2) to find the optimal survival separation point at $t$. Then, we compute this for all time points $t = 1, \ldots, T$ to get the complete survival boundary. Finally, for each of the *test* time series based on its size, we can obtain the survival probabilities. If the estimated survival probability for the test time series is less than $S^*(t)$ at $t$ then the cascade is determined to be viral. Similarly, Eq. (3) can be used to compute the predicted time of virality. The total *time complexity* of SansNet is $O(m(|R| + T))$, where computing all risk intervals takes $O(m|R|)$ and finding the optimal survival boundary takes $O(mT)$, which makes it very scalable.

```
Algorithm SansNet($X = \{x_1, \ldots, x_m\}$: $m$ input time series,
                  $\tau$: threshold)
begin
    $y$ = Get labels for each time series using threshold $\tau$
        and its final size (+1 if above threshold, -1 otherwise)
    $R$ = Get risk time points, where a time series was censored
        or crossed threshold $\tau$.
    Initialize $D$  ##(data table used for Cox-estimation)
    for each $x_i$ in $X$
        $\hat{x}_i$ = Break the time series into risk intervals
            using the time points in $R$
        $d_i$ = Get current size at the end of each risk interval
            for time series $x_i$
        Add $d_i$ to $D$
    end for
    $M$ = Train Cox-extended model using $D$
    $T$ = Maximum risk time point in $R$
    for each $t = 1, \ldots, T$
        $s^*_t$ = Find the optimal $S^*$ value at time $t$ using
            model $M$ and the current size of each time series
            at time $t$.
    end for
return($S^* = s^*_1, \ldots, s^*_T$)
end
```

**Figure 3: SansNet Algorithm**

## 4. EMPIRICAL RESULTS

Predicting large cascades is a hard problem. In this section, we evaluate the effectiveness of the prediction quality and the Early Prediction Advantage (EPA) for all cascades and show that our approach performs well in both measures.

### 4.1 Data Sets

We have used three real-life data sets from two popular social networks, Facebook and Twitter.

**Facebook:** We collected the hourly reshare counts of a random sample of 250,000 public[2] *photos* and *videos* that were uploaded on August 8, 2015. We tracked the hourly reshare counts of these photos and videos for a period of one week.

**Twitter:** In order to evaluate our approach on a public data set we used a Twitter data set[3], which was also used in a recent paper [37]. This data set contains 166,072 tweets and their exact time of reshares and the resharers number of followers. This data set was collected from October 7 to November 7, 2011. The data set was already pre-filtered to contain only tweets that had at least 50 reshares or more. Using this data set also allows us to directly compare our results, as it has been used recently in cascade prediction studies (as described in baselines).

### 4.2 Baselines

To compare SansNet, we use a variety of different types of baselines that cover both recent [37, 29] and popular ones, network-centric and non-network centric approaches including regression and classification schemes.

[2] Public content can be seen by everyone on Facebook.

[3] `http://snap.stanford.edu/seismic/`

**Linear:** We used a linear model as described in [29] for this baseline: $\ln(R_\infty) \sim \ln(\beta \cdot R_t) + \epsilon$, where $R_\infty$ is the final size of the cascade and $R_t$ is the total number of reshares (or retweets) at time $t$. $\epsilon$ is the noise term that accounts for the randomness in the individual content dynamics. Note that the final size and current sizes are log-transformed in this model. For the purpose of classification, if the predicted final size reaches (>=) the threshold, then we consider it to be a viral cascade and not viral otherwise.

**Logistic:** We considered the total number of reshares for cascade $i$ at each hour until the prediction hour $t$ as the feature set for this classifier, i.e. $X^t(i) = [x_i(1), \dots, x_i(t)]$. For each prediction hour $t$, we constructed a separate classifier with this feature set. The training and testing ground truth was constructed based on the actual final size. If the final size reaches the threshold the cascade is classified as viral (+1) and not viral (-1) otherwise. Importantly, as the current total size until time $t$ (i.e. $\sum_{j=1}^{t} x_i(j)$) and hourly rate of change of reshares $(x_i(t) - x_i(t-1))$ are collinear to the actual reshares observed (i.e. feature set $X^t(i)$), we ended up including the actual hourly reshare counts until $t$. This baseline is similar to the one constructed in [4], as the most important features in this cascade prediction model are their *temporal features*.

**CTree:** CTree stands for conditional inference trees [12]. They learn a decision-tree based model on the input feature space and depending on the output variable they can either be regression or classification trees. The main difference between CTree and decision trees is how the variable selection and splitting is performed. In CTree the statistically significant variable with the lowest p-value is selected and the split is performed using the maximum value of the test statistic. Moreover, the stopping criterion is statistically grounded and hence no further pruning is required as needed in decision trees. We use the same set of features used in the Logistic baseline and the output variable is binary—whether the cascade is viral or not based on the threshold.

**Seismic:** This is a very recent approach proposed in [37]. The approach uses a stochastic process model, called the self-exciting point process, to model the spread of the reshares in the network. The self-exciting model assumes that all previous instances influence the future evolution of the process. The authors show the performance gains for this approach over other feature-based baselines. They use network information such as degree of each resharer and also the exact time of reshare of each tweet. The method predicts the final size of the cascade—so if the predicted final size is above the threshold, we predict that cascade as a viral cascade. We used the code available as a R package[3].

We do not use network-based features for our other baselines in part so that we can do a fair comparison. However, we compared against Seismic anyway (which uses the network) on *the same data set as in* [37] primarily to *objectively* test whether network-agnostic approaches like SansNet can get similar (or even better) performances to the best network-aware ones. Note that the comparison is inherently a bit unfair for our approach as by definition SansNet uses less information. Indeed the motivation for our paper is that the underlying network is frequently unavailable or too noisy for use.

## 4.3 Evaluation Setup

We evaluated all approaches using a 3-fold cross validation scheme. The error bars reported are one standard deviation of the three fold evaluation. We set up three different experiments to evaluate the effectiveness of the approach. The virality threshold $\tau$ for both Facebook and Twitter datasets was set at 99.5-th percentile, i.e. 0.05% of cascades go viral.

The first set of experiments are evaluating the virality prediction effectiveness at different life times of the cascades, say the 4th, 8th and 12th hour of the cascade. Though the absolute time of cascade is useful in understanding how effective each classifier is based on the age of the cascade, it does not tell how well the classifier can classify relative to the time of virality. It may be the case that some methods are good at detecting long range signals while some are better at detecting a couple of hours before the cascade goes viral. Hence, to understand this effect we run our second set of experiments to evaluate various metrics relative to the time to virality, say $1, 2, 3$ hours *before* the cascade becomes viral (hits the threshold). The third set of experiments measures the effectiveness of predicting the time *of* virality itself compared to their actual time.

The first two sets of experiments use the standard evaluation measures from classification literature, such as F-measure. F-measure is a harmonic mean of precision and recall that measures the balance between them, using a parameter beta. As our problem is a rare class classification, we use a beta of three. For the third task, we measure the Early Prediction Advantage (EPA) defined in Section 3.4 i.e. the average difference between the actual time of virality (crossing the threshold $\tau$) and first predicted time of virality. So the better performing approach should have a larger EPA. If a method fails to predict the cascade going viral, then the difference is counted as zero (that is the prediction time is same as when the cascade crossed the threshold $\tau$).

## 4.4 Prediction and Cascade Age

How well can we predict the large cascades? We evaluate the effectiveness of all approaches, in Fig. 4, in terms of their absolute age in hours since they have started. We consider exponentially growing time gaps for our analysis, since most of the cascade activity happens at the beginning of the cascade (within the first 24 hours). For each prediction time $t$ we compute the predictions from all approaches after eliminating that cascades that are already viral at that prediction time. This way no approach gets the trivial advantage of predicting things that have already gone viral (i.e. crossed $\tau$). We *do not* use any input filtering threshold for Facebook data set and we considered all data *as-is.* For the publicly available Twitter data set only tweets that have crossed the threshold of 50 reshares are available.

**Results:** As one can see from Fig. 4, SansNet gets a good balance between precision and recall and consistently outperforms all approaches in terms of F-measure. The linear models do really well only in the beginning of the cascade, as they tend to estimate the size really poorly as time progresses. This is particularly because the log of current cascade size is not really predictive enough of the final size, in part due to the highly non-linear nature of the signal. However, the logistic models do very well for older cascades as the complete reshare sequence is made use of in estimating the rates of change. CTree performs extremely poorly as there is no viable decision boundaries that delineates all the viral cascades from the non-viral ones. Also, the non-linear (log) transformations on the input size seems to really make a significant difference in performance.

The most surprising aspect is the performance of Seismic. It performs better than most of the other approaches, especially towards the later times. This is due to the fact that there are only a few cascades that go viral after 24 hours and these cascades that last longer have crossed the 'supercritical' state; so Seismic is able to make robust predictions in their case. However still, the availability of network information to this approach does not improve significantly over the baselines. More importantly, SansNet which does not explicitly use any network information, seems to perform *much better* than Seismic overall. This is mainly because survival probabilities are computed from hazard rates which are robust estimators of instantaneous rates of change. They implicitly account for the network structure in terms of the rise and fall of the reshare time series. For instance, consider the two extreme examples of a chain and clique graph. Now, if we grow a cascade with simple BFS process at each time step, it is easy to see that rate of change of reshare cascades for chain is constant and clique is exponential. Then, the survival functions will reflect this network structure by having a constant slope for chain or larger slope for clique compared to any natural (or power-law) graphs.

## 4.5 Prediction and Time to Virality

How early can we predict the cascade outbreak? We measured the effectiveness of each method at various stages of the growth of cascade in Section 4.4. This gives a perspective on relative performance of each approach with respect to the age of the cascade. However, this does not tell us how effective each approach is *before* the cascade hits the virality threshold. Next, we evaluate the prediction performance of the methods relative to the time at which the cascade crossed the (virality) threshold.

**Time to virality:** We define the term *time to virality* as the number of hours before the hour at which cascade crossed the threshold. For example, if the cascade crossed the threshold at 10th hour after it was first posted, then *time to virality* is 3 hours at the 7th hour for that cascade. Each social network has different dynamics due to the nature of the underlying network (bi- vs. uni-directional) and its purpose (friending vs following). Accordingly, in the datasets we used, the median time to virality for Twitter, Facebook Photos and Videos were 8, 15 and 23 hours respectively. As most of the activity happens with-in *twice the median value*, we focus on this period for all our analysis. Given a time to virality in hour as $k$, we compute the predictions from all methods exactly $k$ hours before it hits the viral threshold. Note that this applies only for cascades that went viral. Cascades that did not go viral will not have a *time of virality* reference point and hence cannot be included in this analysis. Also, note that if a cascade went viral at $k$-th hour (say 10th hour) then time to virality is valid only until $k-1$ (i.e. 9th hour).

We firstly measured recall, the number of cascades that are correctly predicted given the total number of cascades that went viral, in this analysis. Precision or F-measure cannot be computed in this analysis, as cascades that did not go viral do not have virality time as a reference point to compute the *time to virality* (x-axis) for these plots.

In addition, to measure our effectiveness at predicting the top of the list, we also use the Breakout coverage@$k$: it is the fraction of correctly predicted viral cascades out of the top-$k$ cascades (based on its final size at the end of each observation period). It tells us how well each method performs at the top of the list, where the list of viral cascades are sorted in descending order by the final size.

**Results:** See Fig. 5. It is evident from Fig. 5(a,d,g) that all approaches do very well 1 hour before the outbreak, in terms of recall value of ∼0.9 for Facebook and ∼0.8 for Twitter data sets. However, this may be too short for any manual intervention or strategic decision making. As time progresses each method's performance degrades at a different rate. SansNet seems to have the slowest rate of change and tends to predict with considerably high recall (up to ∼60% improvement) at 24 hours before the viral outbreak. Other baselines find it hard to predict the outbreak with a staggering low recall of 0.35 or below for Facebook photo data set. CTree performs the worst amongst all baselines and in contrast linear and logistic approaches are quite competitive.

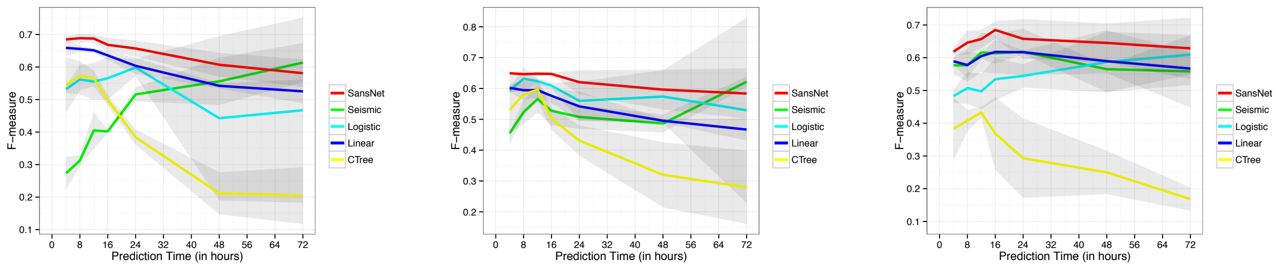
Coverage@10 is trivial for most approaches and our approach overlaps a lot with other baselines. Though, as we increase the top-$k$ to 20 and 30 we begin to see the difference accumulating between SansNet and the baselines. This is clearly visible by comparing the plots in the second and third column of Fig. 5. The complete coverage at 100% of the data is the recall which is shown in the first column of Fig. 5. In the Twitter data set, all approaches have a slower rate of degradation as most of the activity happens in the first 8 hours of the cascade.

## 4.6 Early Prediction Advantage

Above, we have measured the performance of each approach with respect to the actual time of virality of the cascade. Recall and breakout coverage @ k measures how well each approach can recover the cascades that went viral several hours before it reaches its virality threshold. Here we measure how early (on average) each method can predict virality across all viral cascades, called *Early Prediction Advantage* (EPA). Specifically, as defined in Section 3.4, EPA is the mean of *actual time of virality* minus the *prediction time* across all cascades that went viral. The prediction time here is the *first time* the approach predicted that the cascade will go viral. The prediction time for our algorithm SansNet is given by Eq. (3). For other baselines, we sweep the cascade at every time point and pick the first time point at which the approach predicted the cascade as viral. For those cascades that were missed to be classified as viral, we assign a value of 0.

The EPA metric is an important measure for evaluating different approaches for our problem. A method with low recall that could predict only several hours before the actual time of virality, is preferred over a method with high recall, but predicts the viral cascade only a hour before the virality.

**Results:** See Fig. 6. SansNet is a clear winner in all data sets. Seismic does really well in Twitter data set—however, it does not do that well in Facebook data set. One reason could be several parametric assumptions that it makes, including the functional form of the memory kernel $\phi(s)$ [37]. In contrast, SansNet is semi-parametric and the baseline hazard used in our models is completely non parametric, which offers the flexibility to work well for any dataset.

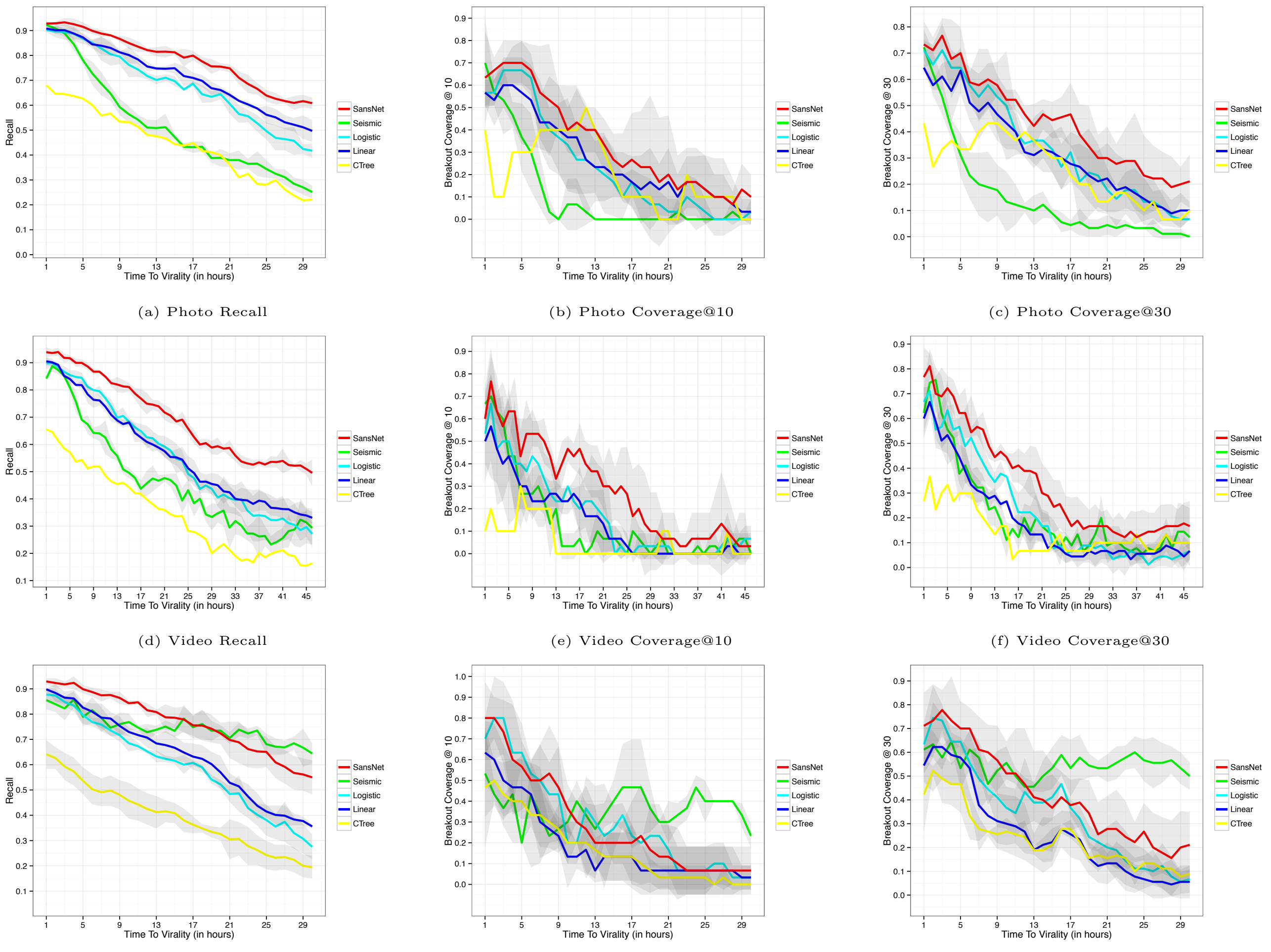


(a) Facebook Photos

(b) Facebook Videos

(c) Twitter

**Figure 4: Prediction effectiveness results for Facebook and Twitter data sets.**



(a) Photo Recall

(b) Photo Coverage@10

(c) Photo Coverage@30

(d) Video Recall

(e) Video Coverage@10

(f) Video Coverage@30

(g) Twitter Recall

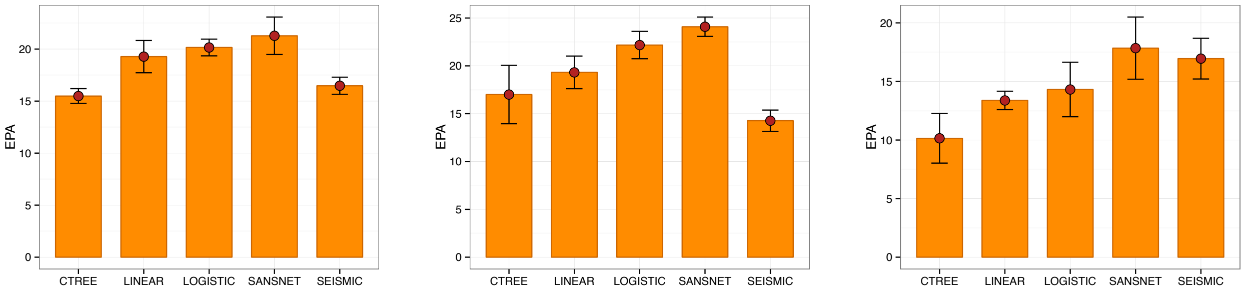
(h) Twitter Coverage@10

(i) Twitter Coverage@30

**Figure 5: The recall and breakout coverage results for time to virality experiments.**

## 4.7 Scalability

Finally, note that running SansNet is really cheap based on the relative time and cost benefits compared to other baselines. As a reference point, SansNet takes only 125.18 seconds for training (and testing is just O(1) for us) on Twitter data set on average per fold. Seismic took 1832.51 seconds and was the slowest baseline as it needs to estimate the parameters and predict for size for each cascade separately. Other baselines ran faster, but their performance in terms of Recall and F-measure was poor (less by upto 60% in terms of recall). In terms of run time, CTree took 55.14, Logistic took 115.02, and Linear Regression took 13.94 seconds. Thus SansNet offers the best tradeoff interms of both efficiency and effectiveness, among all the evaluated methods, for the virality prediction problem.



(a) FB Photo (b) FB Video (c) Twitter

**Figure 6: Early Prediction Advantage (EPA) results.**

## 5. CONCLUSIONS

We address the important problem of detecting a large cascade on social networks in this paper. In contrast to most of the state-of-the-art, the key novelty of our approach SansNet is that it is *network-agnostic*. Using the concept of survival functions from medical statistics, we develop a supervised classifier to estimate the non-linear decision boundary to separate the viral-cascades from non-viral ones. Our results especially show that our network agnostic approach performs very well when the cascade is young and robustly on all datasets unlike other network-aware ones: showcasing its generalizability, and effectiveness on sparse data.

Future work can look into incorporating some content-based features to our method, particularly in the Cox model as additional features, to tailor it more for any domain (like 'diseases' vs 'dog-videos').

## 6. REFERENCES

[1] R. M. Anderson and R. M. May. *Infectious Diseases of Humans*. Oxford University Press, 1991.

[2] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe. Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, 2008.

[3] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[4] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *Proceedings of the 23rd International Conference on World Wide Web*, pages 925–936. ACM, 2014.

[5] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. In *PNAS*, 2008.

[6] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang. Cascading outbreak prediction in networks: a data-driven approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 901–909, 2013.

[7] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *ICWSM*, 2014.

[8] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[9] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.

[10] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML (3)*, pages 666–674, 2013.

[11] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42, 2000.

[12] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[13] K. Kapoor, M. Sun, J. Srivastava, and T. Ye. A hazard based approach to user return time prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1719–1728. ACM, 2014.

[14] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD Iinternational Conference on Knowledge Discovery and Data Mining*, 2003.

[15] J. P. Klein and M. L. Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2003.

[16] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proceedings of 12th International World Wide Web Conference*, pages 568–576, New York, NY, USA, 2003. ACM Press.

[17] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1059–1068. ACM, 2010.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 420–429. ACM, 2007.

[19] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Patterns of cascading behavior in large

blog graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 551–556. SIAM, 2007.

[20] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos. Thermocast: a cyber-physical forecasting model for datacenters. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1370–1378. ACM, 2011.

[21] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[22] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos. Spatio-temporal mining of software adoption & penetration. In *IEEE/ACM Advances of Social Network Analysis and Mining (ASONAM)*, Niagara Falls, CA, Aug 2013.

[23] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos. Fractional immunization in networks. In *Proceedings of SIAM International Conference on Data Mining*, pages 659–667, 2013.

[24] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and Information Systems*, 2012.

[25] B. Ribeiro, M. X. Hoang, and A. K. Singh. Beyond models: Forecasting complex network processes directly from data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 885–895. ACM, 2015.

[26] E. M. Rogers. *Diffusion of Innovations, 5th Edition*. Free Press, August 2003.

[27] K. Subbian, C. Aggarwal, and J. Srivastava. Content-centric flow mining for influence analysis in social streams. In *Proceedings of Conference on Information & Knowledge Management*, pages 841–846. ACM, 2013.

[28] K. Subbian and P. Melville. Supervised rank aggregation for predicting influencers in twitter. In *Proceedings of 3rd IEEE International Conference on Social Computing*, pages 661–665, 2011.

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

[30] T. M. Therneau. *A Package for Survival Analysis in R*, 2015. version 2.38.

[31] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012.

[32] S. Wang, Z. Yan, X. Hu, P. S. Yu, and Z. Li. Burst time prediction in cascades. In *AAAI, January 25-30, 2015, Austin, Texas, USA.*, pages 325–331, 2015.

[33] J. Yang and S. Counts. Predicting the speed, scale, and range of information diffusion in twitter. *ICWSM*, 10:355–358, 2010.

[34] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 599–608, 2010.

[35] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, pages 177–186. ACM, 2011.

[36] L. Yu, P. Cui, F. Wang, C. Song, and S. Yang. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *IEEE 15th International Conference on Data Mining*, 2015.

[37] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1513–1522, 2015.