Non-linear Dynamics of Information Diffusion in Social Networks

YASUKO MATSUBARA, Kumamoto University, JST PRESTO
YASUSHI SAKURAI, Kumamoto University
B. ADITYA PRAKASH, Virginia Tech.
LEI LI, Toutiao Lab
CHRISTOS FALOUTSOS, Carnegie Mellon University

The recent explosion in the adoption of search engines and new media such as blogs and Twitter have facilitated the faster propagation of news and rumors. How quickly does a piece of news spread over these media? How does its popularity diminish over time? Does the rising and falling pattern follow a simple universal law? In this paper, we propose SPIKEM, a concise yet flexible analytical model of the rise and fall patterns of information diffusion. Our model has the following advantages: (a) unification power: it explains earlier empirical observations and generalizes theoretical models including the SI and SIR models. We provide the threshold of the take-off vs. die-out conditions for SPIKEM, and discuss the generality of our model, by applying it to an arbitrary graph topology; (b) practicality: it matches the observed behavior of diverse sets of real data; (c) parsimony: it requires only a handful of parameters; and (d) usefulness: it makes it possible to perform analytic tasks such as forecasting, spotting anomalies, and interpretation by reverse engineering the system parameters of interest (e.g. quality of news, number of interested bloggers, etc.). We also introduce an efficient and effective algorithm for the real-time monitoring of information diffusion, namely, SPIKESTREAM, which identifies multiple diffusion patterns in a large collection of online event streams. Extensive experiments on real datasets demonstrate that SPIKEM accurately and succinctly describes all the patterns of the rise-and-fall spikes in social networks.

Categories and Subject Descriptors: H.2.8 [**Database applications**]: Data mining

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Information diffusion, Social networks, Non-linear modeling

ACM Reference Format:
Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li and Christos Faloutsos, 2017. Non-linear Dynamics of Information Diffusion in Social Networks *ACM Trans. Embedd. Comput. Syst.* 9, 4, Article 39 (March 2010), 41 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Online social media are spreading news and rumors in new ways, and search engines have greatly facilitated this operation, creating bursts and spikes. Some rumors (or memes, hashtags) start slowly and linger; others spike early and then decay; others show more complicated behavior, as we show in Figure 1.

Are there qualitative differences between real rise-and-fall patterns? Do they form different classes? If yes, how many and what kind? Earlier work on YouTube data claims there are four classes [Crane and Sornette 2008]. Empirical work found six

Author's addresses: Yasuko Matsubara, Faculty of Advanced Science and Technology, Kumamoto University, 2-39-1 Kurokami, Chuo-ku, Kumamoto 860-8555, Japan, yasuko@cs.kumamoto-u.ac.jp.

DOI: http://dx.doi.org/10.1145/0000000.0000000

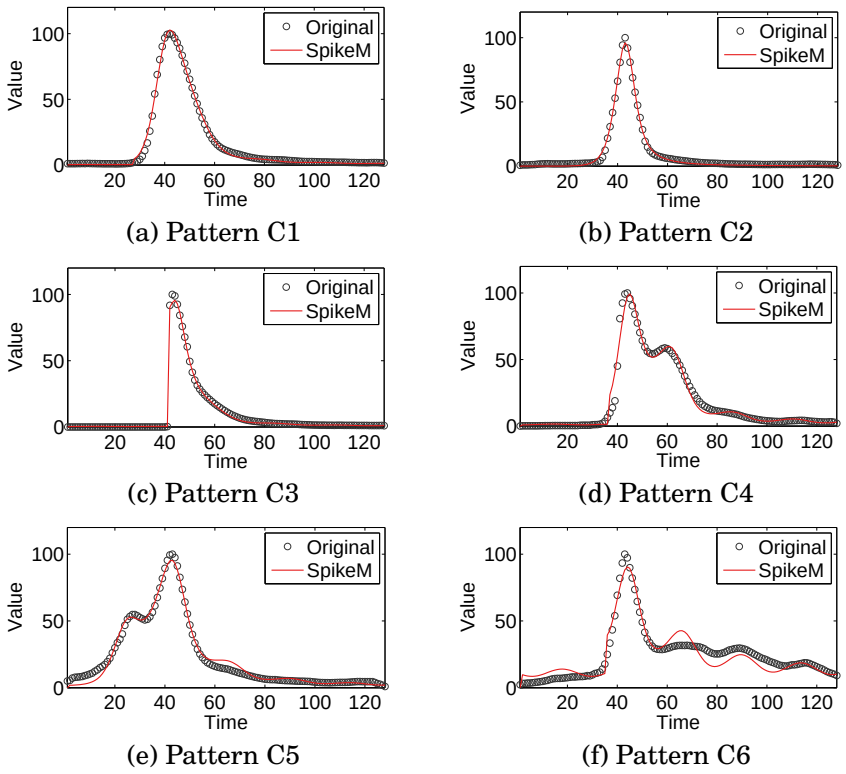


Fig. 1. Modeling power of SPIKEM: six types of spikes (K-SC) shown as dots, and our model fit shown by the solid red line. Data sequences span over 120 time ticks, while SPIKEM requires only seven parameters. The fit is so good, that the red line is often invisible, due to occlusion.

classes [Yang and Leskovec 2011]. How many classes are there after all? —Our answer is: *one*. We provide a non-linear analytical model, SPIKEM, [1] that requires only a handful of parameters, and we show that it can generate all the patterns found in real data simply by changing the parameter values.

Preview of our results. Figure 1 shows six representative spikes of online media (memes) from K-SC [Yang and Leskovec 2011], as gray circles, as well as our fitted model, as a solid red line. Notice that the fitting is very good, despite the fact that our SPIKEM model requires only seven parameters, and that the time-sequences span 120 intervals.

The problem we want to solve is how to model/predict an online activity (e.g., number of blog postings), as a function of time, given some breaking-news at a given time tick. We will use a blogger example for brevity and clarity, but many other processes could be also modeled (such as search volume for popular keywords, rumors spreading over Twitter, and computer viruses infecting machines [Papalexakis et al. 2013]. Consequently, we have:

PROBLEM 1 (WHAT-IF). **Given** *a network of bloggers (/hosts/users), a shock (e.g., event) at time* n_b*, the interest/quality of the event, the count* S_b *of bloggers that imme-*

[1]Available at http://www.cs.kumamoto-u.ac.jp/~yasuko/software.html

Table I. Capabilities of approaches. Only our approach meets all specifications.

	K-SC	C-S	SI/SIR	DWT/DFT	AUTOPLAIT	SARIMA	**SPIKEM**
Domain knowledge	-	√	√	-	-	-	√
Information diffusion	√	√	-	-	-	-	√
Power law decay	-	√	-	-	-	-	√
Non-linear	-	√	√	-	-	-	√
Periodicity	-	-	-	√	-	√	√
Missing values	-	-	√	-	-	-	√
Outlier detection	-	-	√	√	√	-	√
Segmentation	-	-	-	-	√	-	√
Online processing	-	-	-	-	-	-	√
Forecasting	-	-	-	-	-	√	√

diately (= time n_b) blog about the event, **find** *how the blogging activity will evolve over time.*

A closely related problem is to develop a parsimonious model, that can be made to fit several spikes observed in the past (as we do in Figure 1). That is,

PROBLEM 2 (MODEL DESIGN). **Given** *the behavior of several spikes in the past,* **find** *an equation / model that can explain them, with as few parameters as possible.*

More importantly, it would be good if the parameters had an intuitive explanation (such as, 'number of bloggers', 'quality of news', etc, as opposed to, say, a_1, a_2 of an autoregressive model (AR/ARIMA)).

1.1. Contrast with competitors

Table I illustrates the relative advantages of our method. Only SPIKEM matches all requirements.

The six clusters of rise-and-fall patterns in K-SC [Yang and Leskovec 2011] (shown in Figure 1) are non-parametric, and are incapable of forecasting. The C-S method [Crane and Sornette 2008] can capture power law decay patterns, but cannot generate exponential growing patterns or periodic user activities.

The Susceptible-Infected (SI) model and its variations (e.g., SIR, SIRS, SEIR models) are non-linear models, and lead to exponential decay, as opposed to the power law decay that we observe in real data (we will discuss this later in Figure 2). The logistic function [Brauer and Castillo-Chavez 2001], WTA [Prakash et al. 2012], and others equations [Jackson 1992; Nowak 2006; Matsubara et al. 2014b] are non-linear dynamical systems, and the Bass model [Bass 1969] (i.e., the market penetration of new products), the product life cycle model [Klepper 1996; Chang et al. 2014], the input-output model [Leontief 1986] and other related economic models incorporate domain knowledge. However, these methods are not intended to capture long heavy-tail patterns, or periodic user activities.

Wavelets (DWT) and Fourier transforms (DFT) and other basic tools of time-series analysis can detect bursts and typical patterns, but they cannot detect non-linear evolutions of information diffusion in social networks. AutoPlait [Matsubara et al. 2014], SWAB [Keogh et al. 2001] and pHMM [Wang et al. 2011] have the ability to capture the dynamics of sequences and perform segmentation, however, they are not intended to capture long-range non-linear evolutions of information diffusion.

All the traditional, linear time-series methods are *fundamentally unsuitable*: AR, ARIMA, SARIMA and derivatives including AWSOM [Papadimitriou et al. 2003], TBATS [Livera et al. 2011], PLiF [Li et al. 2010] and TriMine [Matsubara et al. 2012a] are all based on *linear* equations, and are thus incapable of modeling data governed by non-linear equations. They lead to exponen-

tial decays, as opposed to the power law that reality seems to obey, and they cannot incorporate domain knowledge. We should also note that all these linear models can go to *infinity* over time.

Our proposed model, SPIKEM is able to successfully replicate the earlier patterns, and also easily fit several, diverse, real datasets. It is very intuitive, and based on diffusion and influence propagation. Moreover, being a concise model, it provides all the related benefits: it can achieve compression, anomaly detection, and is also capable of forecasting.

1.2. Contributions

In this paper, we propose a unifying model, namely, SPIKEM, to solve both the aforementioned problems. Our model has the following advantages:

(1) **Unification power**: it includes earlier patterns and models as special cases (e.g., the SI and SIR models, as well as the patterns in K-SC [Yang and Leskovec 2011; Leskovec et al. 2009]). Our model can also be generalized to an arbitrary graph topology, as well as a clique network.
(2) **Practicality**: it matches the behavior of numerous, diverse, real datasets, including power law decay.
(3) **Parsimony**: it requires only a handful of parameters.
(4) **Usefulness**: our proposed model makes it possible to answer 'what-if' questions (see subsection 6.1), spot outliers, reverse-engineer the system parameters (quality of news, count of interested bloggers, time-of-day behavior of bloggers). We also provide a scalable algorithm, namely, SPIKESTREAM, which is designed for the real-time monitoring of information diffusion (see subsection 4.4).

Our model is made possible by a careful design that incorporates (a) the power law decay in infectivity, (b) a finite population, and (c) proper periodicities. Earlier models cannot handle one or more of the above issues. Thanks to the *practicality* of SPIKEM, we can achieve forecasting, the analysis of 'what-if' scenarios, and the detection of the diffusion spikes and anomalies, as we show in section 5 and section 6.

1.3. Outline

The rest of the paper is organized as follows: Section 2 presents an overview of related work and Section 3 describes the proposed model. In Section 4, we analyze our model, and discuss the generality and extensions of SPIKEM. Sections 5 and 6 show our experimental results for a variety of datasets. We describe related work in Section 7 and conclude this paper in section 8.

2. BACKGROUND

This section describes the fundamental concepts.

2.1. Epidemiology fundamentals

The most basic epidemic model is the 'Susceptible-Infected' (SI) model. Each object/node is in one of two states - Susceptible (S) or Infected (I). Each infected node attempts to infect each of its neighbors independently with probability β, which reflects the strength of the virus. Once infected, each node stays infected forever. If we assume that the underlying network is a clique of N nodes, and use our notation ('B' for blogged = infected) the most basic form of the model is:

$$\frac{dB(t)}{dt} = \beta * (N - B(t))B(t) \tag{1}$$

where the time t is considered continuous, dB/dt is the derivative, and the initial condition reflects an external shock (say, $B(0) = b$ externally infected people).

The justification is as follows: β is the strength of the virus, that is, the probability that an encounter between an infected person ('B') and an uninfected one will result in an infection - and we have $B * (N - B)$ such encounters. The solution for $B()$ is the sigmoid, and its derivative is symmetric around the peak, with an exponential rise and an exponential fall (we discuss this later in Figure 2). There we also show the weakness of the SI model: real data have a power-law 'fall' pattern.

2.2. Self-exciting Hawkes process

Crane et al. [Crane and Sornette 2008] used a self-exciting Hawkes conditional Poisson process [Hawkes and Oakes 1974] to model YouTube views per day, showing that spikes in the activity have a power law rise pattern, and a power law fall pattern, depending on the model parameters. Roughly, the Hawkes process is a Poisson process where the instantaneous rate is not constant but depends on the count of previous events, whose effect drops with the age τ of the event. That is, if there are a lot of events (viewings/bloggings) recently, we will have many such events today.

The base model states that the rate of spread of infection depends on (a) the external source $S(t)$ and (b) self-excitation, that is, on earlier-infected nodes ($i = 1, \ldots$); these nodes spread the infection with decaying virus strength $\phi(\tau)$, their age τ grows, times some constant μ_i. The constant μ_i is equivalent to the degree of the infected node i.

$$\frac{dB(t)}{dt} = S(t) + \sum_{i, t_i \leq t} \mu_i \phi(t - t_i) \tag{2}$$

The model typically assumes that the μ_i values are equal, namely that all nodes have the same degree ('homogeneous' graph). Under certain conditions, the model provides power-law rise and power-law fall patterns.

Next we present our proposed model, SPIKEM, which avoids the shortcomings of the SI and Hawkes models, and has several other desirable properties.

3. PROPOSED METHOD

In this section, we provide the reader with several interesting and important observations, and present our proposed model, namely, SPIKEM. For simplicity, we first focus on the most basic case: a clique network, where all nodes (i.e., bloggers) are potentially connected to each other with undirected and unweighted edges.

3.1. Design philosophy of SPIKEM

Basically, our model tries to capture the following behaviors, which we observed for several of our real data:

— P1: power-law fall pattern
— P2: periodicities

and at the same time we want to

— P3: avoid the divergence to infinity

that other models may have. To handle P3 (divergence), we force our model to have a finite population, and adjust the equations accordingly. To handle P1 (power-law fall pattern), we assume that the infectivity of a node (= popularity of a blog post) decays with the influence exponent p. The handling of periodicities is discussed in subsection 3.3. We describe our model in steps of increasing complexity, and we start with the base model.

Table II. Symbols and definitions

Symbol	Definition
N	Total population of available bloggers
n_d	Duration of sequence
n	Time tick ($n = 0, \ldots, n_d$)
$U(n)$	Count of **u**ninformed bloggers
$B(n)$	Count of informed **b**loggers
$\Delta B(n)$	Delta: count of newly informed **b**loggers at time n
$f(\tau)$	In**f**ectiveness of a blog-post, at age τ
β	Strength of infection
$S(n)$	Volume of external **s**hock at time n
n_b	Starting time of **b**reaking news
S_b	Strength of external shock at birth (time n_b)
ϵ	Background noise
P_a	Strength of periodicity
P_p	Period (e.g., $P_p = 24$ hours)
P_s	Phase shift of periodicity

We assume there are N bloggers, and none of them is yet blogging about the topic of interest. At time n_b, an event occurs (such as the 2004 Indonesian tsunami, or a controversial political speech such as 'lipstick on a pig'), and S_b bloggers immediately blog about it. We refer to this external event as a *shock*, and n_b and S_b are the birth time and the initial magnitude of the shock.

Our model needs a few more parameters: the first is the quality/interestingness of the news, which we denote as β, since this is the standard symbol for the infectivity of a virus in epidemiology literature. If β is zero, nobody cares about this specific piece of news; the higher the value, the more bloggers will blog about it.

Finally, we have the decay function $f(\tau)$, which models how infective/influential a blog posting is, at age τ. Standard epidemiology models assume that $f()$ is constant (once sick, you have the same probability of infecting others); recent analysis has shown that the influence drops with age, following a power law.

The above are the parameters of the base model. Before we list the equations, we want to briefly mention a derived quantity, $\beta * N$; this quantity roughly corresponds to the R_0 ('R-naught') found in the epidemiology literature. This tells us the size of the "first burst": if only one person was infected, how many will be infected in the next time tick?[2]

In summary, the scenario we model is as follows:

— nothing happens, until a news-event appears, at birth time n_b.
— S_b bloggers immediately blog about it.
— other bloggers visit the initial S_b (or follow-up) bloggers, and occasionally get 'infected' and blog about the event, too.

We also assume that

— each blogger blogs at most once about the event
— no other related event occurs - that is, the shock function $S()$ has only one spike.

Without loss of generality, we also assume that once an uninformed blogger sees an infected/informed blog, he/she always blogs about the event (if he/she blogs with probability $\rho < 1$, we could absorb ρ in the infectivity factor β).

Our goal is to find an equation to describe the number $\Delta B(n)$ of people blogging at time tick n, as a function of n and of course the system parameters (total number

[2]yes, it should be $N - 1$, but we sacrifice accuracy, for intuition.

of bloggers N, strength of infection β, etc). Table II lists the major symbols and their definitions.

3.2. Base model - SPIKEM-BASE

The model we propose has nodes (=bloggers) of two states:

— U: **U**ninformed of the rumor
— B: informed, and **B**logged about it

For those who were just informed at time tick n, we will use the symbol $\Delta B(n)$, and we assume that, once informed, a person will blog about the rumor immediately.

Let $U(n)$ be the number of uninformed people at time n, and let $\Delta B(n)$ the number of people who just found out about the rumor at time n, and blogged about it immediately.

MODEL 1 (SPIKEM-BASE). *Our base model is governed by the equations*

$$\Delta B(n+1) = U(n) \cdot \sum_{t=n_b}^{n} \big(\Delta B(t) + S(t)\big) \cdot f(n+1-t) + \epsilon \tag{3}$$

$$U(n+1) = U(n) - \Delta B(n+1) \tag{4}$$

where

$$f(\tau) = \beta \cdot \tau^{-p} \tag{5}$$

and initial conditions:

$$\Delta B(0) = 0, \;\; U(0) = N$$

In addition, we add an external shock $S(n)$, a spike generated at birth time n_b. Mathematically, it is defined as follows:

$$S(n) = \begin{cases} 0 & (n \neq n_b) \\ S_b & (n = n_b) \end{cases} \tag{6}$$

Justification of the model. We undertake this in steps:

— The term $\Delta B(t) + S(t)$ captures the number of bloggers plus external sources, that were activated at time tick t; their infectivity is modulated by the $f()$ infectivity function, since we assume that the infectivity of a source/blogger decays with time. The summation is over all past time ticks since the birth time n_b of the shock.
— The infectivity function $f()$ exactly follows a power law with exponent p. We set $p = 1.5$ as discovered by earlier work on read data: real bloggers [Leskovec et al. 2007b], and responses to mails by Einstein and Darwin [Barabasi 2005].
— The meaning of the summation is the available stimuli at time tick n; the available targets are the uninformed bloggers $U(n)$, and the product gives the number of new infections.
— We add a noise term ϵ to handle cases such as the meme 'yes we can'; some bloggers mention this phrase anyway, but a large shock occurred during the 2008 political campaign, (i.e., it was a slogan for Barack Obama). Very often, $\epsilon \simeq 0$.

This completes the justification of our base model.

We also mention some rules that our model obeys. By definition,

$$B(n) = \sum_{t=0}^{n} \Delta B(t)$$

and of course we have the invariant

$$B(n) + U(n) = N$$

where N is the total number of people/bloggers.

3.3. With periodicity - SPIKEM

Bloggers may modulate their activity following a daily cycle (or weekly, or yearly). For example, a fraction of the $U(n)$ uninformed bloggers at time n are not paying attention (say, because they are tired or asleep). So, how can we reflect this in our equations? We propose an answer below, and then we provide the justification.

MODEL 2 (SPIKEM). *We can capture the periodic behavior of bloggers with the following equations:*

$$\Delta B(n+1) = p(n+1) \cdot \Big(U(n) \cdot \sum_{t=n_b}^{n} \big(\Delta B(t) + S(t)\big) \cdot f(n+1-t) + \epsilon \Big) \tag{7}$$

$$p(n) \ = 1 - \tfrac{1}{2} P_a \Big(sin\big(\tfrac{2\pi}{P_p}(n + P_s)\big) + 1 \Big) \tag{8}$$

where $U(n)$, $S(t)$ and $f(n)$ are defined in Model 1.

Justification. The model is identical to SPIKEM-BASE, with the addition of a periodicity factor $p(\cdot)$. This captures the fact that bloggers tone down their activity, e.g., during the night, or even stop it altogether. The idea is that $U(\cdot)$ is the count of victims available for infection, and the summation is the number of attacks. Under normal circumstances, each victim-attack pair would lead to a new victim; however, since the victims are not paying full attention (tired/asleep), the attacks are not so successful, and thus we prorate them by the $p()$ periodic function.

— P_p stands for the period of the cycle (say, 24 hours).
— P_s stands for the phase shift: if the peak activity is at noon, and the period is P_p=24 hours, then P_s=18.
— P_a depends on the amplitude of the fluctuation, and specifically it gives the relative value of the off-time (say, midnight), versus peak time (say, noon). Thus, if P_a=0, we have no fluctuation.

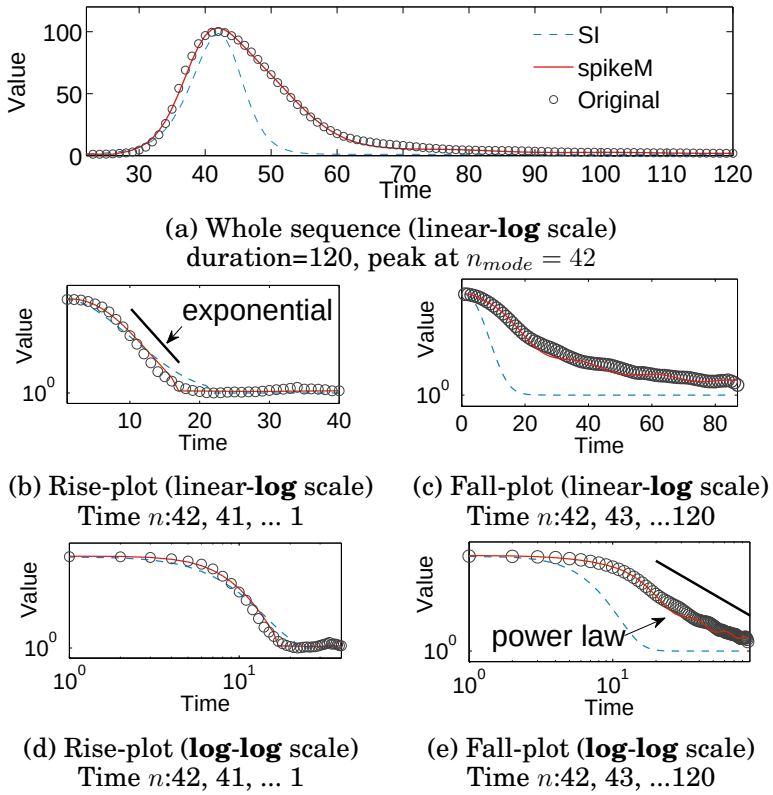
3.4. Analysis - exponential rise and power law fall

Figure 2 shows the behavior analysis result of SPIKEM for Pattern C1 in Figure 1. Specifically, it shows the original time-series data (shown as gray circles), and the fitting results of SPIKEM (red line) and SI (blue dashed line). We can observe that its rise pattern is exponential, while the fall pattern obeys a power law. This is desirable, because this behavior seem to prevail in real data. Let n_{mode} denote the time tick at which the wave $\Delta B()$ reached its maximum volume (that is, $n_{mode} = \arg\max_n \Delta B(n)$).

By *rise plot* we mean the plot of values from the birth time n_b until n_{mode} (and reversing time $abs(n - n_{mode})$) The *fall-plot* is defined similarly: activity $\Delta B()$ versus delay from the peak $n - n_{mode}$. As shown in Figure 2, there is a power law for the fall part, and an exponential shape for the rise part. On the other hand, the traditional SI model, which, as expected, exhibits exponential behavior for both the rise and fall parts.

3.5. Learning the parameters

Our model consists of a set of seven parameters: $\boldsymbol{\theta} = \{N, \beta, n_b, S_b, \epsilon, P_a, P_s\}$. Given a real time sequence $X(n)$ of bloggers at time tick n ($n = 1, \ldots, n_d$), we use the



(a) Whole sequence (linear-**log** scale) duration=120, peak at $n_{mode} = 42$

(b) Rise-plot (linear-**log** scale) Time n:42, 41, ... 1

(c) Fall-plot (linear-**log** scale) Time n:42, 43, ...120

(d) Rise-plot (**log-log** scale) Time n:42, 41, ... 1

(e) Fall-plot (**log-log** scale) Time n:42, 43, ...120

Fig. 2. Fitting results of SPIKEM vs. SI for Pattern C1 in Figure 1. The original sequence (in gray circles), and our model (red line) have an exponential rise part and a power law drop; The SI model (blue dashed line) is exponential for both parts and thus unrealistic. Top row: full interval; left column: only the rise part; right column: only the fall part.

Levenberg-Marquardt (LM) method [Levenberg 1944] to minimize the sum of the errors. The LM algorithm can solve the problem of minimizing a non-linear function in the least square sense. To learn the model parameter set $\boldsymbol{\theta}$, we minimize the prediction error measured by the Euclidean distance between the original and predicted volumes of activity, i.e.,

$$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} D(X, \boldsymbol{\theta}), \qquad D(X, \boldsymbol{\theta}) = \sum_{n=1}^{n_d} (X(n) - \Delta B(n))^2 \tag{9}$$

where, X is the original sequence of duration n_d, and $\Delta B(n)$ is the estimated count of infections at time n given a set of parameters, $\boldsymbol{\theta}$.

4. MODEL ANALYSIS AND EXTENSIONS

In this section, we theoretically analyze our proposed model and provide several important observations and extensions.

4.1. Generality of SPIKEM

As we mentioned in the introduction section, one of the most important properties of SPIKEM is the unification power. Specifically, SPIKEM (i.e.,SPIKEM-BASE) includes several basic non-linear epidemiological models (e.g., SI and SIR) as special cases.

The idea is that we change the infection probability $f(\tau) = \beta * \tau^{-p}$ with the time-since-infection τ. For example, a typical susceptible-infected (SI) model has a constant transmission (i.e., infection) rate β for every time tick, and then, all nodes will eventually become infected. Consequently, we have:

LEMMA 4.1. SPIKEM *is identical to the SI model, if the influence exponent* $p = 0$*, where we have a constant transmission probability over time, i.e.,* $f(\tau) = \beta \cdot constant$.

The susceptible-infected-recovered (SIR) model has an infection rate β and a healing rate δ, each of which describes the transition probability of each state (i.e., from susceptible to infected, and from infected to recovered). More specifically, the healing rate δ defines the constant probability of healings per time tick, which every infected node is exposed to. For example, if $\delta = 0$, no one will recover, and the model has a constant transmission rate β for every time tick, i.e., it is identical to the SI model. If $\delta = 1$, each infected node will recover immediately, that is, the model has a single pulse transmission $\beta \cdot pulse(1)$ for each node, and it is identical to our infectivity function with the exponent $p = \infty$, i.e., $f(\tau) = \beta \cdot \tau^{-\infty} = \beta \cdot pulse(1)$.

LEMMA 4.2. SPIKEM *exhibits the same behavior as the SIR model, if the influence exponent* $p = \infty$ *and the healing rate* $\delta = 1$*, where we have a single pulse transmission at time tick* $\tau = 1$*, i.e.,* $f(\tau) = \beta \cdot pulse(1)$.

4.2. Threshold condition for SPIKEM

Given a social network and a brand new rumor (e.g., a newly released movie), can we determine whether the rumor will *take off* or *die out* quickly? That is, given a new, unknown rumor, how can we guess whether the whole community will be instantly thrown into an uproar, or just ignore it as meaningless information?

We now provide the threshold of the *take off* vs. *die out* conditions for SPIKEM.

THEOREM 4.3 (SPIKEM TAKE-OFF CONDITION). *Given a network of* N *bloggers and the infectivity decay function:* $f(\tau) = \beta \cdot \tau^{-p}$ *with exponent* p $(p < 1)$*, where* β *is the strength of the infection,* SPIKEM *will take off, if it satisfies the following condition:*

$$s = N\beta \cdot \zeta(p) \geq 1.0 \tag{10}$$

where, $\zeta(p)$ *is the Riemann zeta function.*

PROOF. Consider that one person/blogger was infected at time $\tau = 0$. At time $\tau = 1$ (i.e., the first burst), this blogger infects $N\beta \cdot 1^{-p}$ neighboring bloggers. Similarly, at time $\tau = 2$ he/she infects $N\beta \cdot 2^{-p}$ bloggers [3].

Consequently, the total number s of bloggers who are infected by the first blogger is,

$$s = N\beta \cdot 1^{-p} + N\beta \cdot 2^{-p} + N\beta \cdot 3^{-p} + \cdots + N\beta \cdot \tau^{-p} + \ldots \tag{11}$$

That is, summing up all the above counts, we have

$$s = N\beta \cdot \sum_{\tau=1}^{\infty} \tau^{-p} = N\beta \cdot \zeta(p) \tag{12}$$

[3] More specifically, it is $(N - N\beta)\beta \cdot 2^{-p}$ at time $\tau = 2$, but we can discard $O(\beta^2)$ terms when $\beta \ll 1$.

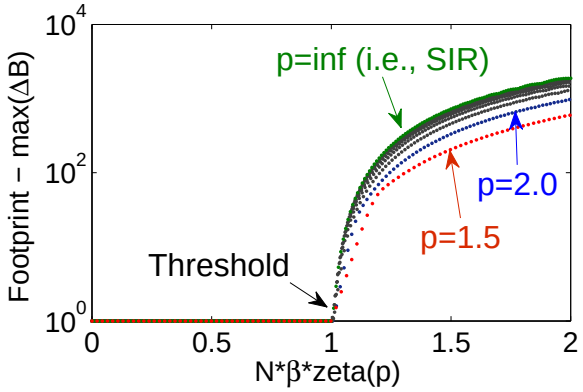


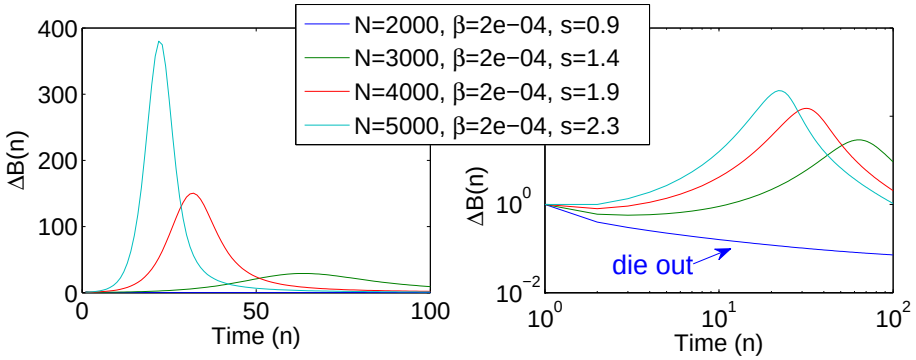
Fig. 3. Illustration of the SPIKEM take-off condition: it shows a scatter plot of our threshold (i.e., $N\beta \cdot \zeta(p)$) vs. the *footprints* (i.e., the maximum value of each spike, i.e., $\max \Delta B(n)$). We vary the condition $0 \leq N\beta \cdot \zeta(p) \leq 2$, with the influence exponent $p = 1.5, 2.0, \ldots, \infty$. Note that each point corresponds to each spike, and the tipping point exactly matches our condition: $s = N\beta \cdot \zeta(p) = 1.0$.

where, $\zeta(p)$ is the Riemann zeta function, i.e., $\zeta(p) = \sum_{\tau=1}^{\infty} \frac{1}{\tau^p}$. Here, if s is less than 1.0, this means that the first blogger cannot infect enough (i.e., more than 1) people in his/her community, thus the news/rumor dies out without receiving any attention. Similarly, if there are multiple infected bloggers at time $\tau = 0$, each blogger needs to infect more than 1 neighbor (i.e., $s \geq 1.0$) to satisfy the take-off condition. □

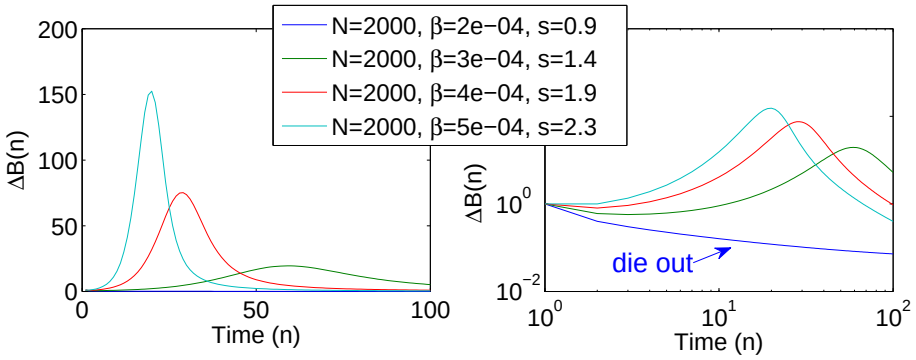
Behavior analysis. Figure 3 shows the threshold analysis simulation result. It shows the scatter plot of the *threshold* vs. *footprints*, that is, the take-off condition (i.e., $s = N\beta \cdot \zeta(p)$) vs. the peak position of each spike (i.e., $\max \Delta B(n)$)). We vary the infection rate β with the fixed population $N = 10^4$ so that we have the condition $0 \leq s \leq 2$, with several influence exponents $p = 1.5, 2.0, \ldots, \infty$. For example, the red points correspond to the spikes with the slope $p = 1.5$. In Figure 3, as we expected, the footprint of the infection in all spikes suddenly jumps at $s = 1$.

We should also note that this condition covers the condition of the basic SIR model. It is well known that the traditional SIR model has an epidemic threshold $N\beta/\delta \geq 1$ [Hethcote 2000]. In Figure 3, the green points show the footprints with the influence exponent $p = \infty$ (here, $\zeta(\infty) = 1.0$), which is equivalent to the SIR model with a healing rate $\delta = 1.0$. Also note that the SI model has no inherent epidemic threshold as all nodes will eventually become infected.

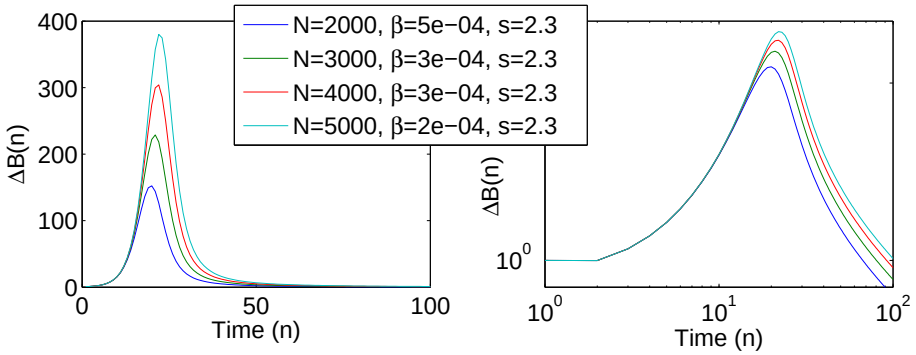
In Figure 4, we present several results for specific parameter settings ($N = 2000$ or so, $\beta = 2 \cdot 10^{-4}$ or so). The figure shows linear-linear (left column) and log-log (right column) scales. We fixed the remaining parameters, i.e., $n_b = 0, \epsilon = 0, P_a = 0, B(0) = 1, p = 1.5$. Figure 4 (a) shows the behavior of SPIKEM, where we vary the total population N from 2000 to 5000, with a fixed infection strength β, while Figure 4 (b) shows the result for $\beta = \{2 \cdot 10^-4, \ldots, 5 \cdot 10^{-4}\}$, with a fixed population $N = 2000$. It should be noted that SPIKEM always takes off, if the condition holds, (that is, $s \geq 1.0$), otherwise, it dies out very quickly, as shown by the blue arrows in the figure (a) and (b). Figure 4 (c) shows another special case, where we vary both parameters N and β, so that we have the fixed condition $s = 2.3$.



(a) $N = \{2000, \ldots, 5000\}$, $\beta = 2 \cdot 10^{-4}$, $s = \{0.9, \ldots, 2.3\}$



(b) $N = 2000$, $\beta = \{2 \cdot 10^{-4}, \ldots, 5 \cdot 10^{-4}\}$, $s = \{0.9, \ldots, 2.3\}$



(c) $N = \{2000, \ldots, 5000\}, \beta = \{5 \cdot 10^{-4}, \ldots, 2 \cdot 10^{-4}\}$, $s = 2.3$

Fig. 4. Behavior analysis: several spikes for a specific setting. We varied the total population N from $2,000$ to $5,000$, and the infection rate β from $2 * 10^{-4}$ to $5 \cdot 10^{-4}$, with constant values of $n_b = 0, \epsilon = 0, P_a = 0, B(0) = 1$. We tried (a) varying N with a fixed β, (b) varying β with a fixed N, and (c) varying both N and β so as to fix the condition $s = 2.3$. SPIKEM always *takes off*, if the condition holds (i.e., $s \geq 1$), otherwise, it dies out quickly, as shown by the blue lines in figures (a) and (b).

4.3. Model extenisions - general arbitrary graph

Thus far, we have seen how SPIKEM behaves in a clique network, where all nodes (i.e. bloggers) are potentially connected to all other nodes. The next question is: *given an arbitrary graph of N bloggers, how will the blogging activity evolve over time?*

Let $\mathbf{A}$ be the adjacency matrix of an arbitrary graph of N nodes (i.e., bloggers), and let $\Delta I_i(n)$ be the probability of node i to be infected/informed at time n. We introduce a new model, namely SPIKEM-G, which can describe the dynamics of information diffusion in an arbitrary graph.

MODEL 3 (SPIKEM-G). *We can generate the spike of the bloggers in an arbitrary graph network $\mathbf{A}$ with the following equations:*

$$\Delta I_i(n+1) = (1 - I_i(n)) \cdot \sum_{t=1}^{n} \sum_{j=1}^{N} \big(\mathbf{A}_{ji} \cdot \Delta I_j(t) \cdot f(n+1-t)\big) \tag{13}$$

$$\Delta B(n+1) = \sum_{i=1}^{N} \Delta I_i(n+1) \tag{14}$$

$$U(n+1) = U(n) - \Delta B(n+1) \tag{15}$$

with the initial conditions:

$$\Delta I_{sid}(n_b) = 1.0, \quad U(0) = N$$

where, sid is the index of the starting node(s)/blogger(s).

Justification. We have the following:

— The adjacency matrix $\mathbf{A}$ stands for the connectivity between each node/blogger pair. Here, SPIKEM-G is identical to SPIKEM, if the adjacency matrix $\mathbf{A}$ is the clique (i.e., $^{\forall i,j}\mathbf{A}_{ji} = 1$).
— $I_i(n)$ describes the cumulative probability of node i to be infected at time n, that is, $I_i(n) = \sum_{t=1}^{n} \Delta I_i(t)$, where, $0 \leq \Delta I_i(n) \leq I_i(n) \leq 1$.
— The term $(1 - I_i(n))$ shows the probability of node i that remains uninformed (i.e., available for the infection) at time n.
— The summation $\sum_{t=1}^{n} \sum_{j=1}^{N} \big(\mathbf{A}_{ji} \cdot \Delta I_j(t) \cdot f(n+1-t)\big)$ represents the cumulative stimuli for node i, where we have N nodes/bloggers. Here, the cumulative stimuli shows the strength of the propagation effects from the neighbor nodes at time tick n, and it is set to be $[0, 1]$.
— We can compute $\Delta B(n)$ (the number of bloggers who were just infected at time tick n), by summing up the probability of each node, i.e., $\Delta B(n) = \sum_{i=1}^{N} \Delta I_i(n)$.
— We assume that a new event happened at time tick n_b, and S_b blogger(s) immediately blogged about it. Here, sid is the node/blogger ID, who started blogging at time tick n_b.

4.4. Real-time monitoring of information diffusion

In many Web-based services (such as blogs, news and Twitter), we observe a large collection of activity/event logs at every time tick. For example, Twitter generates millions of event entries (e.g., hashtags) every hour. From this huge collection of online events, web-site owners can monitor daily activity patterns, find bursts or spikes of information diffusion, and predict the subsequent week to aid the design of advertisements.

One big challenge when analyzing these logs is to handle such large volumes of data at a very high logging rate. Moreover, in practice, real-life event streams contain

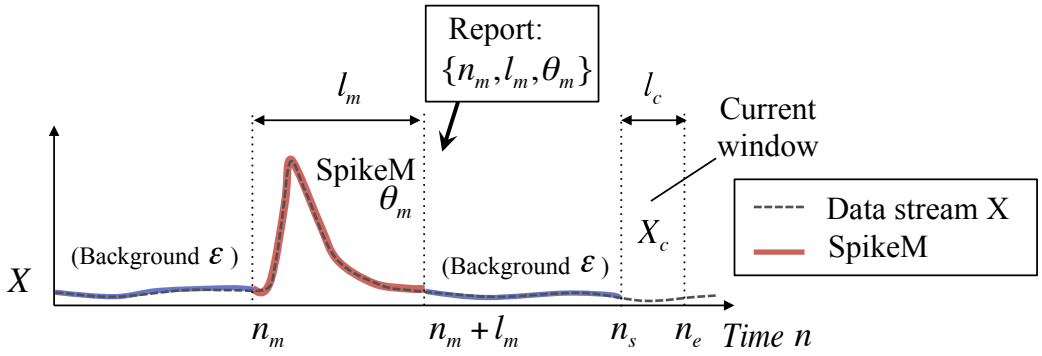


Fig. 5. Illustration of SPIKESTREAM: Given an event stream X, it requires only a single scan to detect the SPIKEM segment(s), and report each cut point (i.e., starting position: n_m, subsequence length: l_m) and model parameter set $\boldsymbol{\theta}_m$. Here, $X_c = X(n_s : n_e)$ is the subsequence of the current window of length l_c.

various types of diffusion patterns of different durations, e.g., multiple spikes for the Harry Potter movie series, as we will see later in Figure 18 (d). That is, we need to identify any sudden discontinuity in an event stream, and recognize the current time-series pattern, immediately, so that we can predict/understand the current diffusion dynamics, adaptively, at any time.

So, how can we deal with this issue? Does our model, SPIKEM help us to solve it? Ideally, we would like to solve the following problem:

PROBLEM 3 (REAL-TIME MONITORING OF INFORMATION DIFFUSION). **Given** *a data stream of online user activities:* $X = \{X(1), X(2), \cdots, X(n), \cdots\}$*, where* n *is the current time tick,* **find** *the segments that have the characteristics of the information diffusion process, incrementally and quickly, that is, we want to*

—identify all subsequences in X *that match the* SPIKEM *model,*
—find cut-points (i.e., starting position n_m *and length* l_m*) of each subsequence,*
—estimate model parameter set $\boldsymbol{\theta}_m$ *for each subsequence.*

Main ideas behind our method. We present a fast, one-path algorithm, namely, SPIKESTREAM. Assume that we have a semi-infinite sequence of activity volume X (e.g., the number of blog-postings/hashtags), i.e., $X = \{X(1), X(2), \cdots, X(n), \cdots\}$, where n is the most recent value. Given a set of newly arriving events for each time tick $1, 2, \cdots, n, \cdots$, our algorithm reports all the qualifying subsequences (i.e., SPIKEM segments), immediately, at any point in time, while discarding redundant information (e.g., background noise). Also note that we might detect very short and *meaningless* spike sequences (say, less than a half-day duration), but this is usually insufficient for many real applications. We thus introduce the minimum length l_c of subsequence matches to enable us to ignore such small subsequences.

Figure 5 illustrates how the algorithm works. Given an event stream X, it extracts the most recently arrived event set, $X_c = X(n_s : n_e)$ of window size l_c. Here, n_s and n_e show the starting and ending positions of the subsequence X_c, and we refer to X_c as a current window. For each disjoint window X_c, SPIKESTREAM tries to fit the SPIKEM model, and it then finds the optimal segment (shown as a red line). Finally, it reports the optimal solution $\{n_m, l_m, \boldsymbol{\theta}_m\}$, (i.e., its starting position, subsequence length, and model paraemter set) in stream processing.

SPIKESTREAM assumes that there are two hidden statuses for each disjoint window in the event stream, i.e.,

Input: (a) a new event $X(n)$ at time tick n and (b) the previous status s_c
Output: a qualifying subsequence $\{n_m, l_m\}$ and its parameter set $\boldsymbol{\theta}_m$ (if any)
if $(n \mod l_c) == 0$ **then**
  /* For each disjoint subsequence X_c of window length l_c */
  $n_s = n - l_c + 1$; // n_s: starting position
  $n_e = n$; // n_e: ending position
  $X_c = X(n_s : n_e)$; // X_c: subsequence of the current window
  /* Calculate the likelihood values of X_c */
  // (1) Treat as background noise, starting from n_s
  $L_\epsilon = \mathcal{N}(X_c|\mu_\epsilon, \sigma_\epsilon^2)$; // $\epsilon = \{\mu_\epsilon, \sigma_\epsilon^2\}$: $\mu_\epsilon = mean(X_c), \sigma_\epsilon^2 = var(X_c)$
  // (2) Treat as a new-born spike, starting from n_s
  $\{\boldsymbol{\theta}, \Delta B\}$ = SPIKEM-FIT ($X_c, \boldsymbol{\theta}^*$) // $\boldsymbol{\theta}^*$: initial SPIKEM parameter set
  $L_c = \mathcal{N}(X_c - \Delta B|\mu_{\boldsymbol{\theta}}, \sigma_{\boldsymbol{\theta}}^2)$; // Likelihood value for $\boldsymbol{\theta}$
  if $s_c ==$ `SpikeM` **then**
    $\{\boldsymbol{\theta}_m, \Delta B\}$ = SPIKEM-FIT ($X(n_m : n_e), \boldsymbol{\theta}_m$); // Model fit using $X(n_m : n_e)$
    $l_m = n_s - n_m$; // l_m: length of the current SPIKEM window
    // (3) X_c belongs to the continuing spike $\boldsymbol{\theta}_m$, starting from n_m
    $L_m = \mathcal{N}(X_c - \Delta B(l_m : l_m + l_c)|\mu_{\boldsymbol{\theta}_m}, \sigma_{\boldsymbol{\theta}_m}^2)$; // Likelihood value for $\boldsymbol{\theta}_m$
    if $L_c > L_m$ or $L_\epsilon > L_m$ **then**
      // Background noise or new-born spike - terminate the current spike
      Report $\{n_m, l_m, \boldsymbol{\theta}_m\}$; // Report the optimal subsequence
      if $L_\epsilon > L_c$ **then**
        $s_c =$`background_noise`; // Switch to background noise
      else
        $n_m = n_s$; $\boldsymbol{\theta}_m = \boldsymbol{\theta}$; // New-born spike - switch to `SpikeM`
      end if
    end if
  else
    if $L_c > L_\epsilon$ **then**
      // New-born spike - switch to `SpikeM`
      $s_c =$`SpikeM`; $n_m = n_s$; $\boldsymbol{\theta}_m = \boldsymbol{\theta}$;
    end if
  end if
end if

Algorithm 1: SpikeStream

(a) `background_noise`: independent activity trend (e.g., random noise or short spikes of less than l_c duration, shown as blue lines in Figure 5). We treat this status as a Gaussian distribution ϵ (i.e., $\mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$).
(b) `SpikeM`: a subsequence/segment that has the characteristics of SPIKEM (i.e., a word-of-mouth phenomenon, shown as a red line in Figure 5).

If the current window X_c belongs to `SpikeM`, the algorithm keeps the starting position n_m of the current subsequence (i.e., $n_m \leq n_s$). If the current window status switches from `SpikeM` to `background_noise`, or, there is a new-born spike, starting at n_s, it reports $\{n_m, l_m, \boldsymbol{\theta}_m\}$ as the optimal subsequence.

Proposed algorithm. Algorithm 1 describes the overall procedure. For each incoming event $X(n)$, it first creates a disjoint subsequence X_c of window length l_c. It then computes the likelihood values of X_c with respect to the following three conditions: (1) L_ϵ: The current subsequence X_c is treated as background noise (i.e.,

$L_\epsilon = \mathcal{N}(X_c|\mu_\epsilon, \sigma_\epsilon^2)$); (2) L_c: There is a new-born spike $\boldsymbol{\theta}$, starting from n_s; (3) L_m: The subsequence X_c belongs to the continuing spike $\boldsymbol{\theta}_m$, starting from n_m. For each condition, we use a Gaussian distribution to compute the likelihood value of X_c. It then determines the optimal condition (i.e., finds the maximum likelihood) so that we obtain the best segmentation. More specifically, if the previous disjoint window belongs to `SpikeM` and the algorithm detects the ending position of the SPIKEM segment, (that is, if the current status s_c switches from `SpikeM` to `background_noise`, or it finds a new-born spike $\boldsymbol{\theta}$ starting at n_s), it reports $\{n_m, l_m, \boldsymbol{\theta}_m\}$, i.e., the starting position n_m, length l_m and the parameter set $\boldsymbol{\theta}_m$, as the optimal solution.

Complexity. Let n be the event stream length and l_m be the maximum length of the qualifying subsequences.

LEMMA 4.4. SPIKEM-OFFLINE *requires* $O(n^2)$ *time and* $O(n)$ *space per time tick.*

PROOF. SPIKEM requires $O(n^2)$ time and $O(n)$ space to calculate the activity volume of length n, i.e., $\{\Delta B(1), \cdots \Delta B(n)\}$ (see Equation 3 in Model 1). □

LEMMA 4.5. SPIKESTREAM *requires at least* $O(1)$ *and at most* $O({l_m}^2)$ *time and at least* $O(1)$ *and at most* $O(l_m)$ *space per time tick.*

PROOF. If the current status is the background noise, SPIKESTREAM requires $O({l_c}^2)$ time and $O(l_c)$ space to compute the new SPIKEM parameter using a current window X_c of length l_c. If the current status is SPIKEM, it needs to update the current SPIKEM parameter set using $X(n_m : n_e)$, where, the length of $X(n_m : n_e)$ is at most $l_m + l_c$. Here, since l_c is a small constant value compared with l_m and n, the complexity can be simplified to $O(1) \sim O({l_m}^2)$ time and $O(1) \sim O(l_m)$ space. □

5. EXPERIMENTS

To evaluate the effectiveness of SPIKEM, we carried out experiments on real datasets. The experiments were designed to answer the following questions:

— Q1: Can we explain the cluster centers of K-SC?
— Q2: How well does our model match *MemeTracker* data?
— Q3: How well does it fit other data?
— Q4: How well does it forecast future patterns?
— Q5: How does it behave in an arbitrary graph?
— Q6: How well does it capture information diffusion patterns in real event streams?

Dataset description. We performed experiments on the following three real datasets.

— *MemeTracker*: This dataset covers three months of blog activity from August 1 to October 31 2008[4]. It contains short quoted textual phrases ("memes"), each of which consists of the number of mentions over time. We choose 1,000 phrases in blogs with the highest volume in a 7-day window around their peak volume.
— *Twitter*: We used more than 7 million Twitter[5] posts of 20 million users covering an 8-month period from June 2011 to January 2012. We selected the 10,000 most frequently used hashtags in a one-week window around their peak volume, with 100,000 users that mentioned these items most frequently.

[4]http://memetracker.org/
[5]http://twitter.com/

Table III. The model parameters of our SPIKEM best fitting on six patterns of K-SC (see Figure 1). Note that the total populations N are around $2,000 - 3,000$, and the strength of the infection $\beta * N = 0.8 - 1.0$ for each pattern (also see the text for details). We see that Pattern C3 has a big exogenous shock at $n_b = 40$, and Patterns C4, C5 and C6 exhibit daily periodicity ($P_a \simeq 0.4$).

	C1	C2	C3	C4	C5	C6
N	2407	1283	1466	3079	4183	3435
$\beta * N$	0.95	1.00	0.86	0.92	0.79	0.69
n_b	26	17	**40**	35	0	34
S_b	4.73	0.06	**114.13**	23.24	2.58	45.58
ϵ	0.36	0.01	0.43	1.48	0.32	13.97
P_a	0.18	0.06	0.22	**0.38**	**0.28**	**0.39**
P_s	12	5	7	**6**	**2**	**2**

— *Google*: This dataset consists of the volume of searches for various queries (i.e., words) on Google[6]. Each query represents search volumes related to keywords over time.

5.1. Q1: Explaining K-SC clusters

The results for this dataset were presented in section 1 (see Figure 1). Our model correctly captures the six patterns of K-SC.

Model analysis. Table III gives a further description of the SPIKEM fitting. Our model consists of seven parameters, each of which describes the behavior of the spikes. Note that the total populations N are almost the same for all patterns (around 2,000 to 3,000). This is because these six patterns are scaled on the y-axis so that they all have a peak volume of 100. In our model, the strength of the infection is described as $\beta * N$. Specifically, we can see that $\beta * N$ is between $0.7 - 1.0$ for these six patterns. We also see that Pattern C3 includes an extreme shock $S_b = 114$ at time $n_b = 40$, which means that this spike was strongly affected by an external burst of activity. Actually, it has a sudden peak and relatively rapid relaxation (see Figure 1 (c)). On the other hand, Patterns C4, C5 and C6 have several peaks about 24 hours apart with a strength $P_a \simeq 0.4$.

Model fitting accuracy. We also evaluated our fitting accuracy by using the root mean square error ($RMSE$) between estimated values and real values:

$$RMSE = \sqrt{\frac{1}{n_d}\sum_{n}^{n_d}(X(n) - \Delta B(n))^2}$$

where, $X(n)$ and $\Delta B(n)$ are original and predicted sequences, respectively. We compared SPIKEM with the (a) SI, (b) SIRS and (c) C-S (i.e., a self-exciting Hawkes process with endogenous/exogenous bursts [Crane and Sornette 2008]) models. Here, for each model, we used the LM method to fit the parameter set, and minimize the error between the original and predicted sequences.

Figure 6 shows the fitting accuracy result for six patterns of K-SC. Note that a lower value indicates a better fitting accuracy. SI has symmetric rise-and-fall patterns, while SIRS generates different rise-and-fall slopes. However, as discussed in section 3 (see Figure 2), these models cannot model the power-law tail parts of the spikes. C-S has the ability to describe power-law growth (i.e., endogenous) or sudden peaks (i.e., exogenous), and it generates power-law relaxation patterns, but cannot generate exponen-

[6]https://www.google.com/trends/

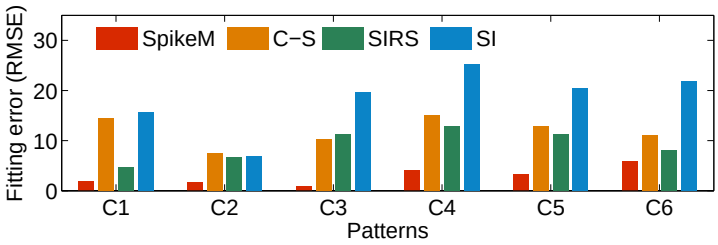


Fig. 6. Fitting accuracy of SPIKEM on six patterns of K-SC (Patterns C1-C6). SPIKEM consistently outperforms its competitors (i.e., C-S, SIRS, SI) with respect to accuracy ($RMSE$) between the original values and the models. Note that a lower value indicates a better fitting accuracy.

tial growing patterns. Also note that, these three competitors cannot describe periodic user activities. On the other hand, our solution, SPIKEM achieves high accuracy for every pattern of K-SC.

5.2. Q2: Matching *MemeTracker* patterns

Figure 7 and Figure 8 show the results of model fitting on the *MemeTracker* dataset. We selected six typical sequences according to the K-SC clusters. That is, each sequence corresponds to each pattern (C1-C6). We show the original sequences (black dots) and SPIKEM fitting, $\Delta B(n)$ (red line) in both linear-linear (top) and log-log (bottom) scales. In the log-log scale, we also show the count of uninformed bloggers, $U(n)$. In Figure 7, the bottom text shows the short phrase (meme) of each sequence. All of the phrases are sourced from U.S. politics in 2008. We obtained several observations for each sequence:

— Patterns C1 and C2: they have almost the same size of population, $N \simeq 500$, except that C2 has a quicker rise and fall (i.e., stronger infection, $\beta * N = 1.4$) than C1 ($\beta * N = 0.94$).
— Pattern C3: this sequence has a sudden rise and a power law decay. There is a slight daily periodicity.
— Patterns C4 and C5: there are clearly daily periodicities. Pattern C6, "lipstick on a pig", has the largest population of all six sequences (i.e., $N = 6259$).
— Pattern C6: the sequence, "yes we can", consists of huge spikes around $n = 40$, and constant periodic noise. This is because the bloggers mention this phrase as Barack Obama's slogan as well as with more general meanings. We can also find that there are several extreme points (i.e., missing values) around $n = 120$ (see blue circle in log-log scale).

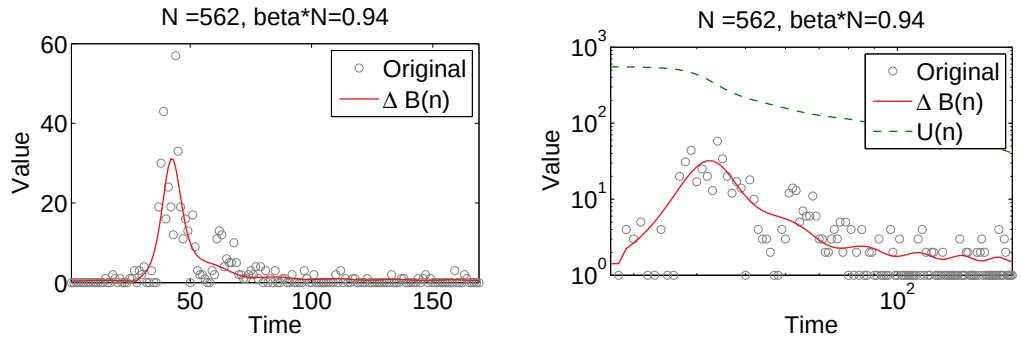
5.3. Q3: Matching other data

We also demonstrate the effectiveness of our model for other types of spikes.

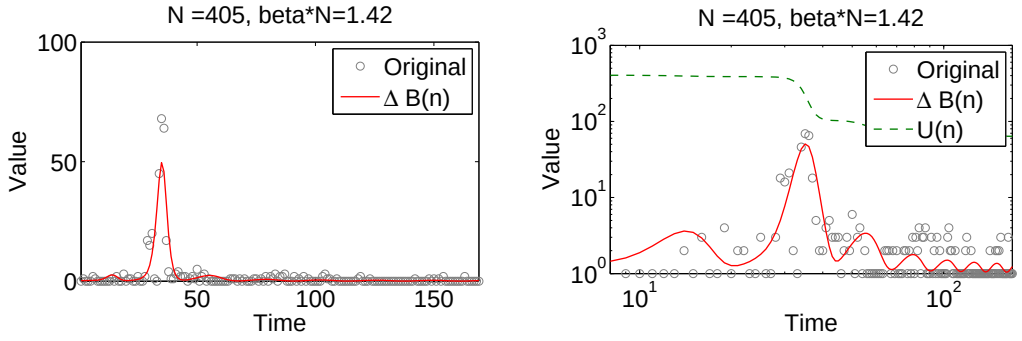
Fitting on Twitter data. Figure 9 and Figure 10 show our fitting results as regards the *hashtags* of *Twitter* data. A hashtag is used to mark keywords or topics in a Tweet, e.g., *#christmas, #newyear*. In these figures, we can see that *Twitter* data behave similarly to *MemeTracker* data (see C1-C6).

Our model captures the following characteristics:

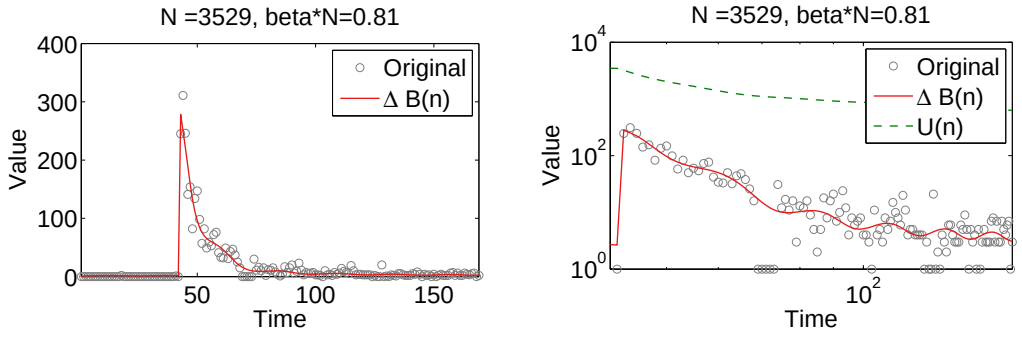
— #assange (Pattern C2): this is a topic about Julian Assange, the founder of WikiLeaks. There are several mentions before the peak point (December 5, 2011).



(a) Pattern C1: Meme #109
"the most serious financial crisis since the great depression"

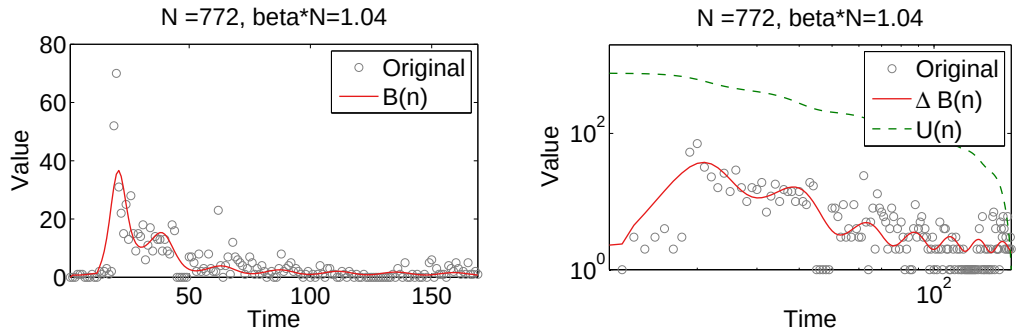


(b) Pattern C2: Meme #34
"i love this country too much to let them take over another election"

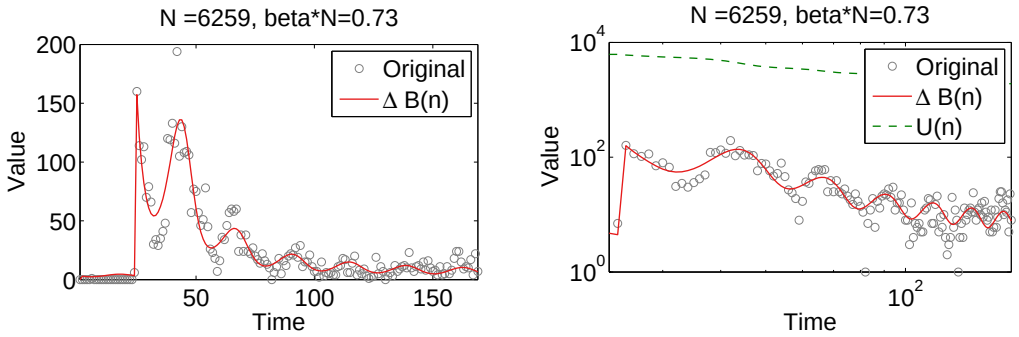


(c) Pattern C3: Meme #13
"hope over fear, unity of purpose over conflict and discord"

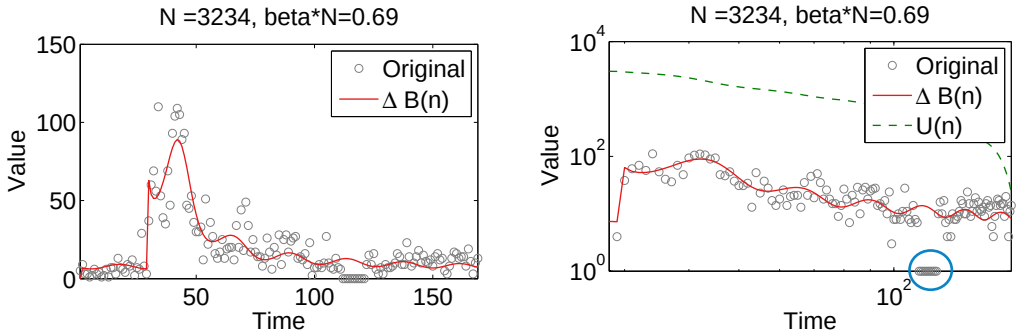
Fig. 7. Fitting results of SPIKEM on six typical patterns from the *MemeTracker* dataset (Pattern C1-C3). The figures are shown in both 'linear-linear' (left) and 'log-log' (right) scales. The bottom text shows the phrase ("meme") of each pattern. We can see that SPIKEM successfully captures each pattern on both linear and log scales.



(d) Pattern C4: Meme #87
"what is required of us now is a new era of responsibility"

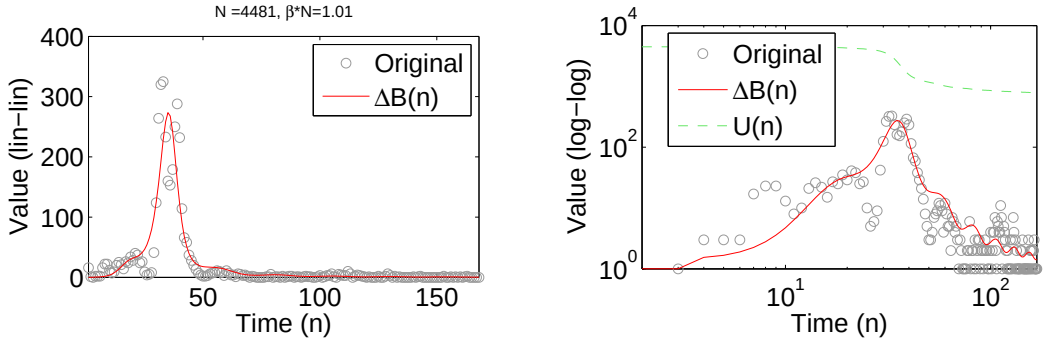


(e) Pattern C5: Meme #9
"you can put lipstick on a pig"

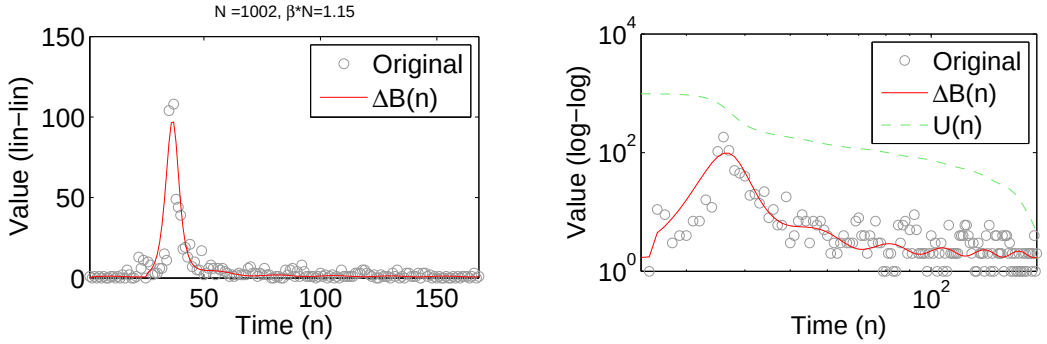


(f) Pattern C6: Meme #3
"yes we can yes we can"

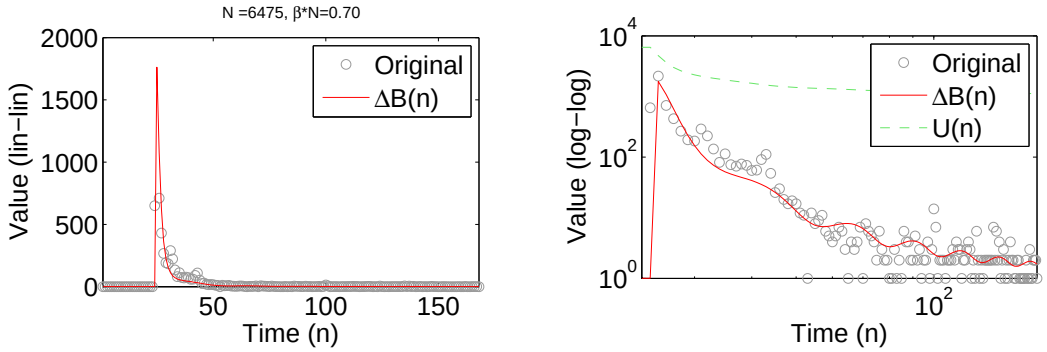
Fig. 8. Fitting results of SPIKEM for six patterns from the *MemeTracker* dataset (Pattern C4-C6). Also note that SPIKEM is robust against noise: we found several extreme points (i.e., missing values) around $n = 120$ in the figure (f) - see blue circle in log-log scale.



(a) Pattern C1 #blacktuesday

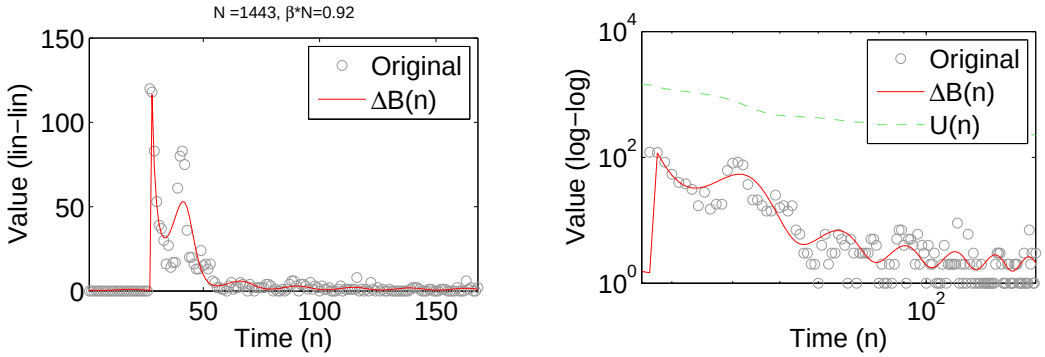


(b) Pattern C2 #assange (Wikileaks)

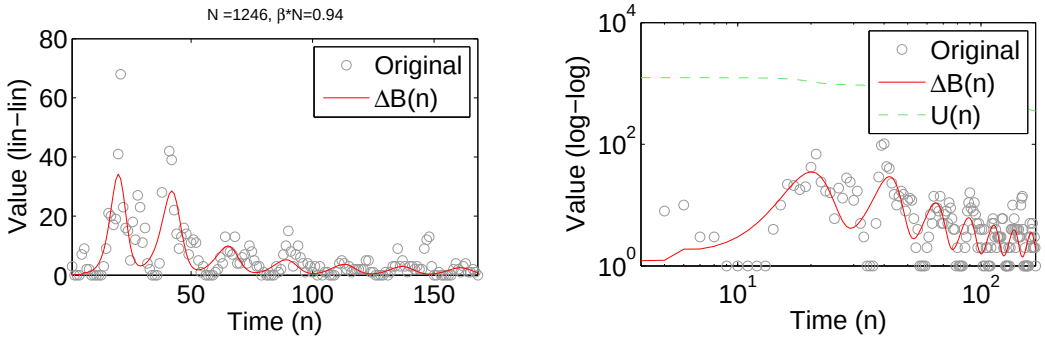


(c) Pattern C3 #ripstevejobs (Rest in peace, Steve Jobs)

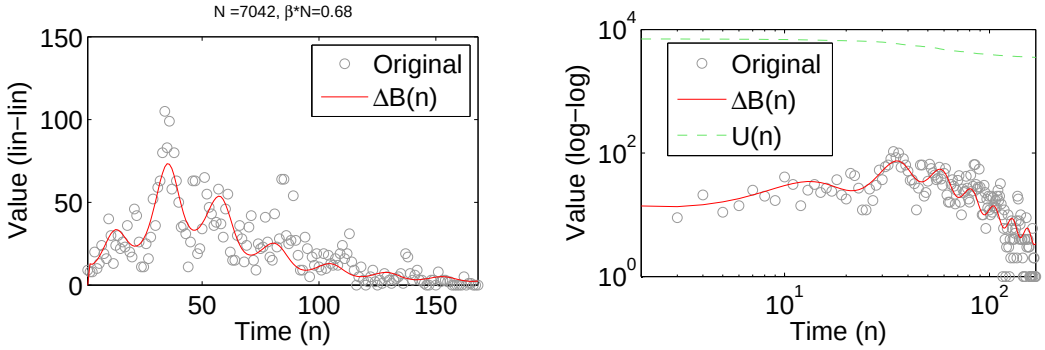
Fig. 9. Fitting results of SPIKEM for six hashtags from the *Twitter* dataset (Pattern C1-C3). The left and right columns show linear-linear and log-log scales, respectively.



(d) Pattern C4 #arresteddevelopment (TV series)

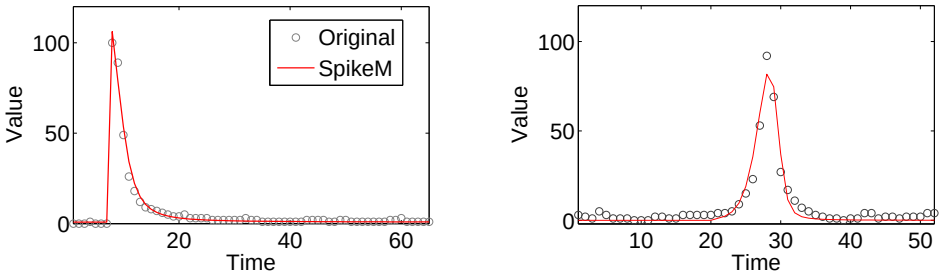


(e) Pattern C5 #ACC (Atlantic Coast Conference)



(f) Pattern C6 #BogorRockStage (music event)

Fig. 10. Fitting results of SPIKEM for six hashtags from the *Twitter* dataset (Pattern C4-C6). The left and right columns show linear-linear and log-log scale, respectively.



(a) "tsunami" (2005) (b) "Harry Potter" (2007)

Fig. 11. SPIKEM fitting for the *Google* dataset: the volume of searches for the keyword (in black dots) and fitting results (in red lines). Note that the window size is per week.

— #ripstevejobs (Pattern C3): there is a sudden peak on October 5, 2011, with a long heavy tail (see Figure 9(b) in log-log scale). This was caused by the death of Steve Jobs (i.e., Rest in Peace, Steve Jobs).
— #arresteddevelopment (Pattern C4): this a topic about the TV series, "Arrested Development". There is a clear daily periodicity with a peak point.

Fitting on GoogleTrend data. We can also observe influence propagation in queries on Internet search engines. Figure 11 shows two different types of spikes on *Google*. Note that this dataset is calculated on a weekly basis, and each volume is scaled so that they all have a peak volume of 100. For an external catastrophic event (a) "tsunami", we see that there is a super quick rise immediately after the Indian Ocean earthquake and tsunami in 2005. In contrast, (b) "harry potter" has a slower rise, which is because this spike was generated by "word-of-mouth" activity surrounding the release of a Harry Potter movie in 2007. SPIKEM successfully captures both types of spikes.

5.4. Q4: Tail part forecasts

So far we have seen how SPIKEM captures the temporal dynamics for various spikes. Here, we answer a more practical question: given the first part of the spike, how can we forecast the future behavior of the tail part? Figure 12 shows our forecasting results for *MemeTracker* data. We selected two phrases with the highest populations (#9 and #13 in Figures 7 and 8). We trained our models by using values obtained over a period of 54 hours (solid black lines in the figure), and then forecasted the following days (solid red lines, about five days). Note that the vertical axis uses a logarithmic scale.

We compared our method with the following forecasting methods: (a) AR, i.e., a traditional forecasting algorithm (for a fair comparison, we used seven regression coefficients with the same size as our model parameters), (b) SARIMA, i.e., seasonal ARIMA (we set $P_p = 24$ hours), where we determined the optimal parameter set using AIC, and (c) TBATS [Livera et al. 2011], i.e., a state-of-the-art forecasting algorithm for complex seasonal time series (we set $P_p = 24$ hours).

As discussed in section 1, AR, SARIMA and TBATS are *unsuitable* for capturing non-linear dynamics; they are *linear* models, and they cannot generate power law decays. Note that, SARIMA and TBATS have the ability to capture sinusoidal cyclic patterns, however, they quickly converge to the zero, or some constant value, and fail to forecast long-range non-linear diffusion patterns.

The right column of Figure 12 shows the forecasting error of each approach (i.e., RMSE between the original and estimated volumes). A lower value indicates a bet-

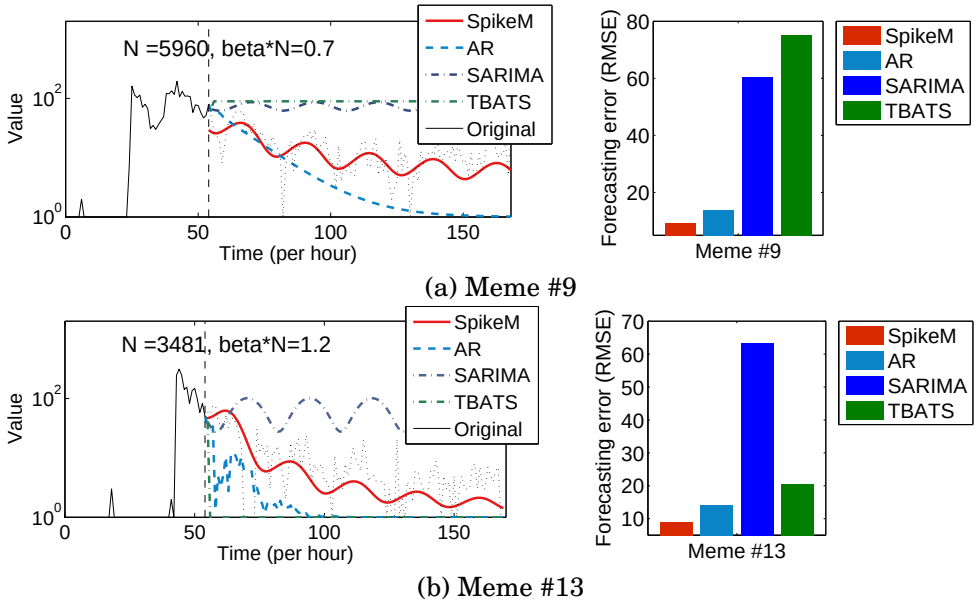


Fig. 12. Results of tail-part forecasting for the *MemeTracker* data. We train spikes from $n = 0$ to 54, and then start forecasting at time $n = 54$. Our SPIKEM better reflects reality, while other methods (i.e., AR, SARIMA, TBATS) cannot capture long-tail decay patterns.

ter forecasting accuracy. Our method achieves a high forecasting accuracy for both sequences.

More importantly, our model can forecast the rise part of spikes as well as the tail part (we discuss this in detail in Section 6).

5.5. Q5: Information diffusion in an arbitrary graph

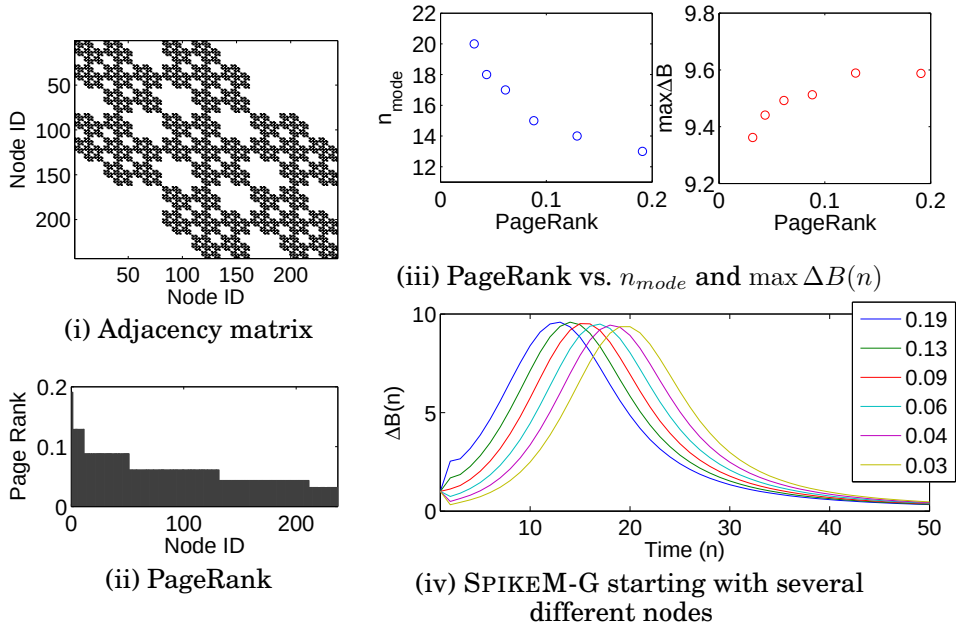
Next, we demonstrate how our proposed model behaves in an arbitrary graph topology. Figure 13, Figure 14 and Figure 15 show our results for Kronecker-Graph [Leskovec et al. 2010], *Twitter* and *Google* [McAuley and Leskovec 2012], respectively [7].

For each figure, the left column shows (i) the adjacency matrix of the given graph, and (ii) the PageRank score of each node. Note that the nodes in figure (ii) are sorted in descending order. The right column shows the results for our model, that is, figure (iii) shows scatter plots of PageRank vs. n_{mode} (left, shown as blue points) and PageRank vs. $\max \Delta B(n)$ (right, shown as red points). In figure (iii), each point represents each trial, starting with a different node (i.e., sid). Figure (iv) shows the spikes of several trials, and the legend box shows the PageRank scores. Note that n_{mode} describes the time tick at which the spike $\Delta B(n)$ reached its maximum value (that is, $n_{mode} = \arg\max_n \Delta B(n)$).

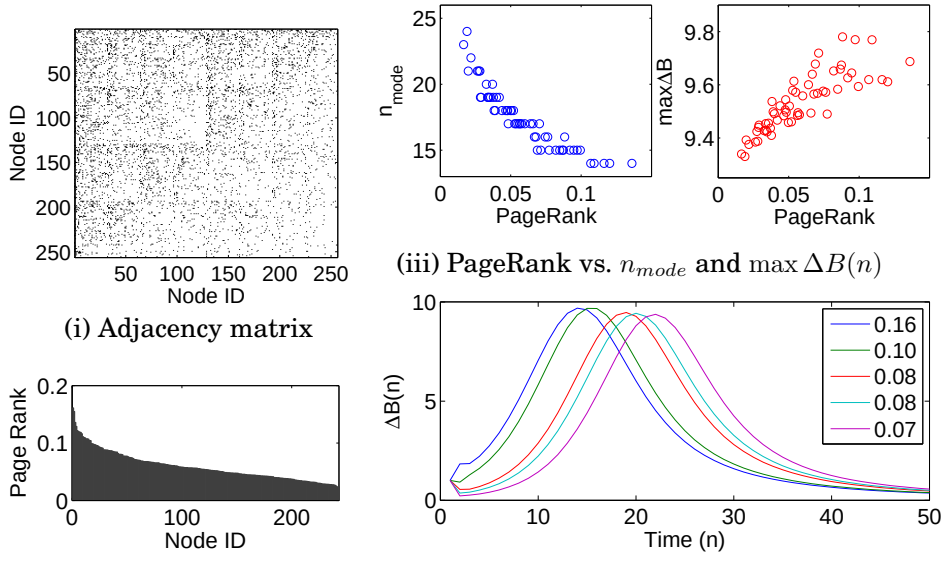
With respect to the SPIKEM-G analysis, we made several interesting and important observations.

OBSERVATION 1 (FAST-RISE). *The PageRank scores of the starting nodes and the time-to-peak values n_{mode} are negatively correlated, for all datasets.*

[7] http://snap.stanford.edu/data/

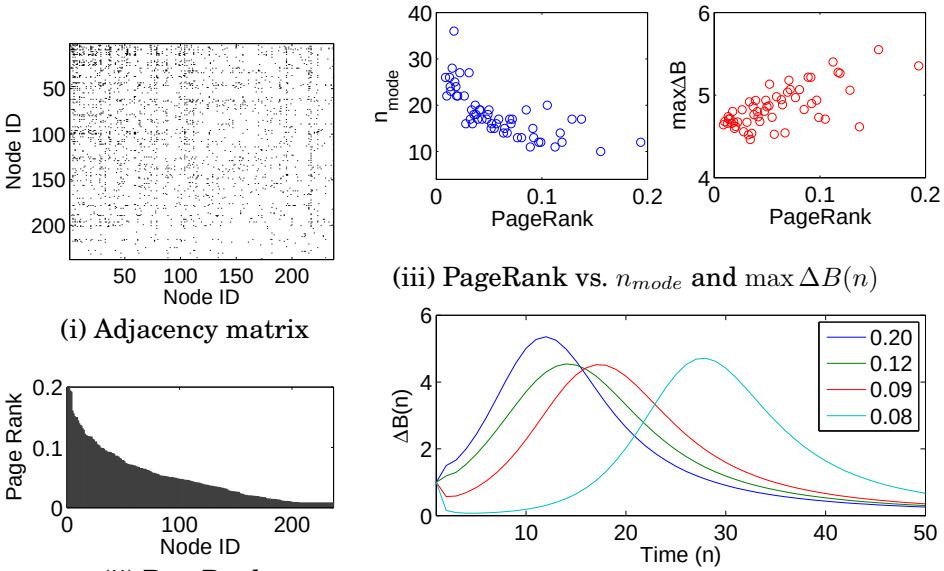


(a) Kronecker graph #1 ($N = 256$)



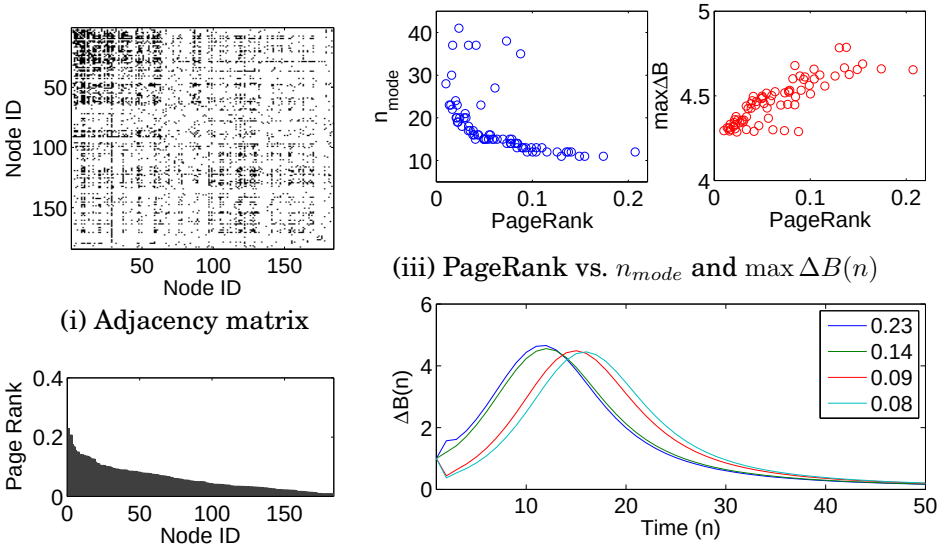
(b) Kronecker graph #2 ($N = 256$)

Fig. 13. Behavior analysis for KroneckerGraph: the left column shows (i) the adjacency matrix and (ii) PageRank of the given graph, while the right column shows our results, that is, (iii) PageRank vs. n_{mode} (left) and PageRank vs. $\max \Delta B(n)$ (right), and (iv) several spikes starting with different nodes (i.e., different PageRank).



(i) Adjacency matrix (ii) PageRank (iii) PageRank vs. n_{mode} and $\max \Delta B(n)$ (iv) SPIKEM-G starting with several different nodes

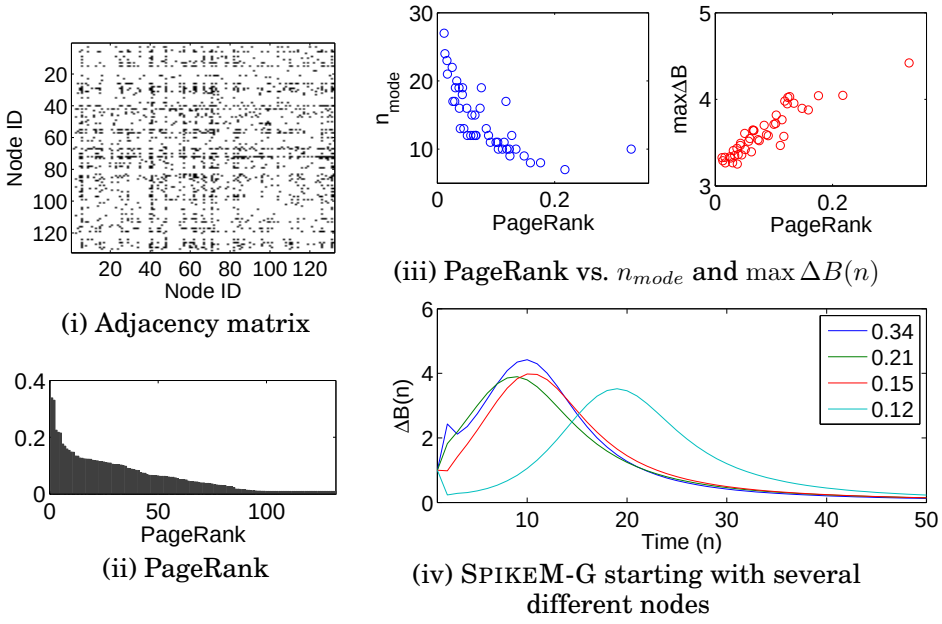
(a) *Twitter* #1 ($N = 237$)



(i) Adjacency matrix (ii) PageRank (iii) PageRank vs. n_{mode} and $\max \Delta B(n)$ (iv) SPIKEM-G starting with several different nodes

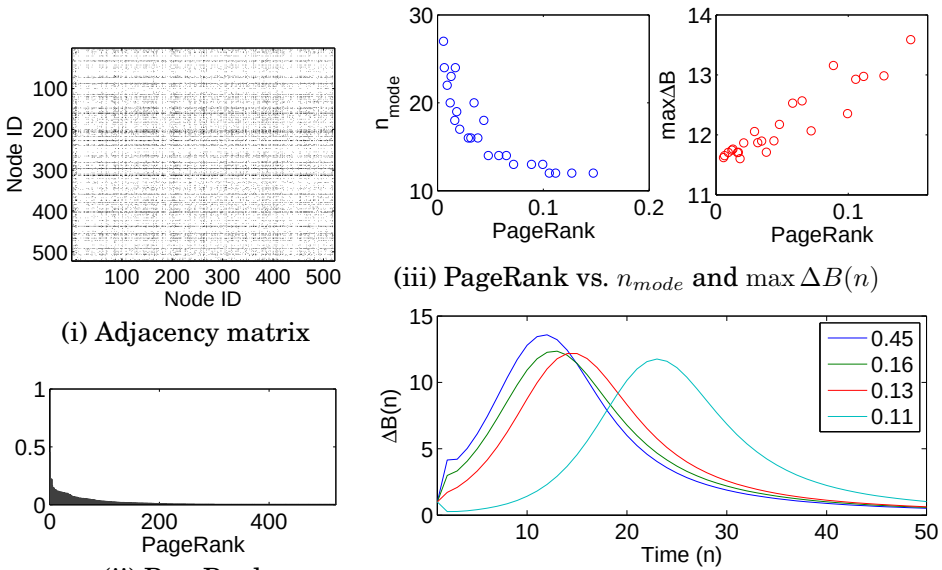
(b) *Twitter* #2 ($N = 184$)

Fig. 14. Behavior analysis for *Twitter*: the left column shows (i) the adjacency matrix and (ii) PageRank of the given graph, while the right column shows our results, that is, (iii) PageRank vs. n_{mode} (left) and PageRank vs. $\max \Delta B(n)$ (right), and (iv) several spikes starting with different nodes (i.e., different PageRank).



(i) Adjacency matrix (ii) PageRank (iii) PageRank vs. n_{mode} and $\max \Delta B(n)$ (iv) SPIKEM-G starting with several different nodes

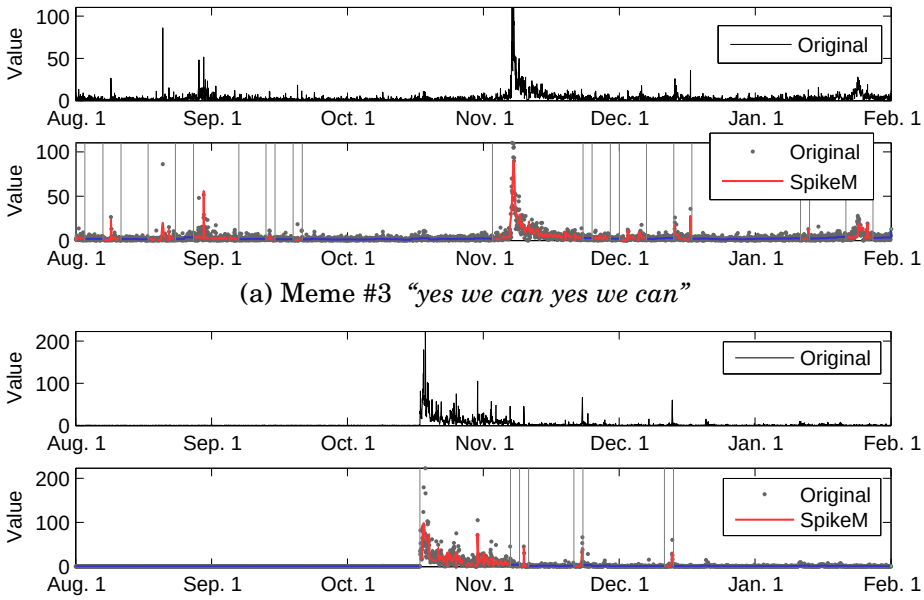
(a) *Google* #1 ($N = 132$)



(i) Adjacency matrix (ii) PageRank (iii) PageRank vs. n_{mode} and $\max \Delta B(n)$ (iv) SPIKEM-G starting with several different nodes

(b) *Google* #2 ($N = 522$)

Fig. 15. Behavior analysis for *Google*: the left column shows (i) the adjacency matrix and (ii) PageRank of the given graph, while the right column shows our results, that is, (iii) PageRank vs. n_{mode} (left) and PageRank vs. $\max \Delta B(n)$ (right), and (iv) several spikes starting with different nodes (i.e., different PageRank).



(a) Meme #3 *"yes we can yes we can"*

(b) Meme #4 *"joe the plumber"*

Fig. 16. Real-time monitoring of information diffusion in the *MemeTracker* streams. The top figure shows the original event stream, and the bottom figure shows our discoveries. Note that the optimal diffusion segments are shown as red lines.

If the starting blogger is strong and popular, that is, he/she has a high PageRank score, a new event/rumor will propagate very quickly through the network. In contrast, if the starting node has a lower connectivity, the model tends to have a long-term propagation process (please see, e.g., the lightblue spike in Figure 14 (a)-(iv)).

OBSERVATION 2 (HIGH-PEAK). *The PageRank scores and the maximum (i.e., peak) points* $\max \Delta B(n)$ *have a weak positive correlation, but no clear difference.*

Compared with the previous plots (i.e., PageRank vs. n_{mode}), there is no clear distinction here. For example, in Figure 15 (b), which has a total population $N = 522$, the peak values $\max \Delta B(n)$ are almost the same for all trials, i.e., between 11-14 (please see the red points in (b)-(iii)).

5.6. Q6: Pattern discovery in real event streams

We now describe how our algorithm effectively and efficiently discovers important patterns and trends in real event streams.

Discovery of information diffusion. SPIKESTREAM discovered the following interesting diffusion patterns:

— *MemeTracker*: Figure 16 shows our results for *MemeTracker* streams (i.e., (a) Meme #3 *"yes we can yes we can"* and (b) Meme #4 *"joe the plumber"*). Each stream consists of blogging activities covering over 6 months, starting from August 1 2008 (on an hourly basis). As shown in the figure, our proposed algorithm identified all the important diffusion spikes (shown as red lines), as well as the positions of the all cut points (shown as vertical lines).

— *Twitter*: Figure 17 shows the results for *Twitter* (hashtags) event streams (starting from August 1, 2008 on an hourly basis). The first three event streams are related to popular TV programs (i.e., *"BONES"*, *"Big Bang Theory"*, *"Breaking Bad"*), where there are clear weekly cyclic spikes, each of which corresponds to the broadcast of a new episode. For example, as shown in figure (c), there are strong weekly spikes and these spikes continue to grow significantly until October 9. In fact, Breaking Bad was one of the most popular TV shows in the U.S., and the final episode of Season 4 was broadcast on October 9. Similarly, SPIKESTREAM can also identify long-range diffusion patterns (as shown in figure (d) *"Hurricane"*), as well as non-periodic multiple spikes (e.g., (e) *"Boxing"*).
— *Google*: Figure 18 shows our results for *Google* streams, which consist of keyword-search volumes covering over ten years (from 2004 to the present, on a weekly basis). SPIKESTREAM captures important trends and the influence propagation process in various types of data streams, such as political terms, e.g., ((a) *"Barak Obama"* and (b) *"Obama care"*), popular keywords, e.g., (c) *"Olympic"* and (d) *"Harry Potter"* and economic crisis ((e) *"Subprime"*).

Scalability. Figure 19 compares SPIKESTREAM with SPIKEM-OFFLINE in terms of computation time for varying sequence lengths n. Note that the figures are shown in log-log scales. As we expected, SPIKESTREAM determines the qualifying subsequences and their model parameters significantly faster than the offline algorithm for large datasets (i.e., up to several orders of magnitude).

6. DISCUSSION - SPIKEM AT WORK

Our proposed model, SPIKEM is capable of various applications. Here, we describe important applications and show some usefulness examples of our approach.

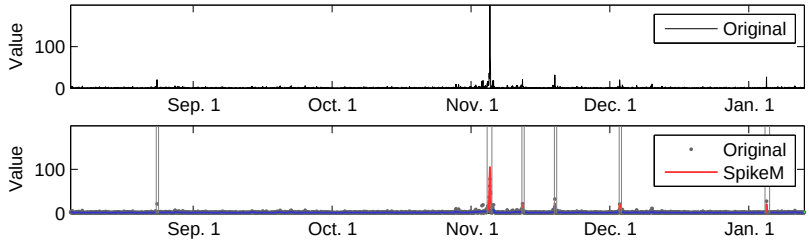
6.1. "What-if" forecasting

We discussed tail-part forecasting in subsection 5.4. Ideally, we want to forecast not only the tail part, but also the rise part of a spike. This is much more difficult, because we usually have very few points in the rise part of a spike. However, if this is a repeating event, such as, say, the spikes induced by the release of 'Harry Potter' movies, can we forecast future spikes if we know the release date of the next movie? It transpires that our SPIKEM model can also help with this (difficult) task.

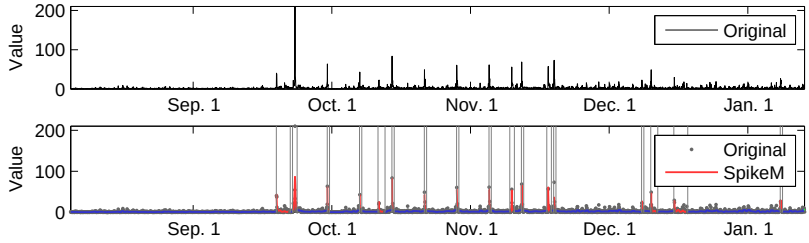
Thus, the problem we address in Figure 20 is as follows: we are given (a) the first spike in 2009, "Harry Potter and the Half-Blood Prince" ($n = 185$); (b) the release dates of the two sequel movies (blue text with arrows pointed at $n = 255$ and 289), and (c) the access volume before the release dates (and specifically from 8 to 2 weeks in advance). Can we forecast the rise and fall shapes of upcoming spikes and their peak points?

Solution and results. SPIKEM can predict the potential population N of users who are interested in "Harry Potter", and the strength of 'word-of-mouth' infection: β. Our solution is to assume that these values are fixed for all subsequent spikes. The only difference is the strength of the "external shock", i.e., n_b and S_b. Our solution consists of the following three-step process:

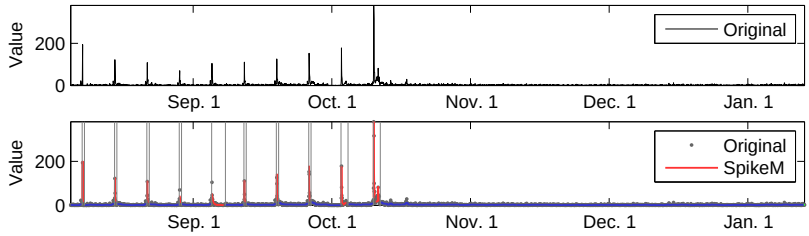
(1) Train the parameter set $\boldsymbol{\theta}$ by using the first spike (solid black line in the figure).
(2) With the fixed parameters $\boldsymbol{\theta}$, infer the new values of $\tilde{n_b}$ and $\tilde{S_b}$ by using the beginning part of the next spike (blue lines between double arrows at $n = 250$ and 280).
(3) Generate the spikes using $\boldsymbol{\theta}$ and $\tilde{n_b}$ and $\tilde{S_b}$ (red lines).



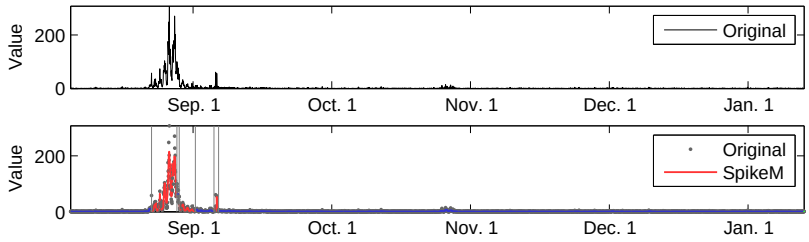
(a) *"Bones"*



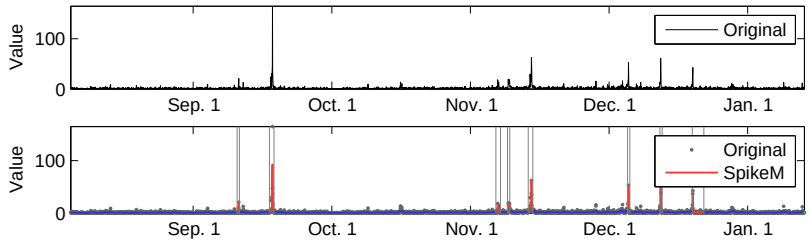
(b) *"Big Bang Theory"*



(c) *"Breaking Bad"*



(d) *"Hurricane"*



(e) *"Boxing"*

Fig. 17. Real-time monitoring of information diffusion in the *Twitter* streams. The top figure shows the original event stream, and the bottom figure shows our discoveries. Note that the optimal diffusion segments are shown as red lines.

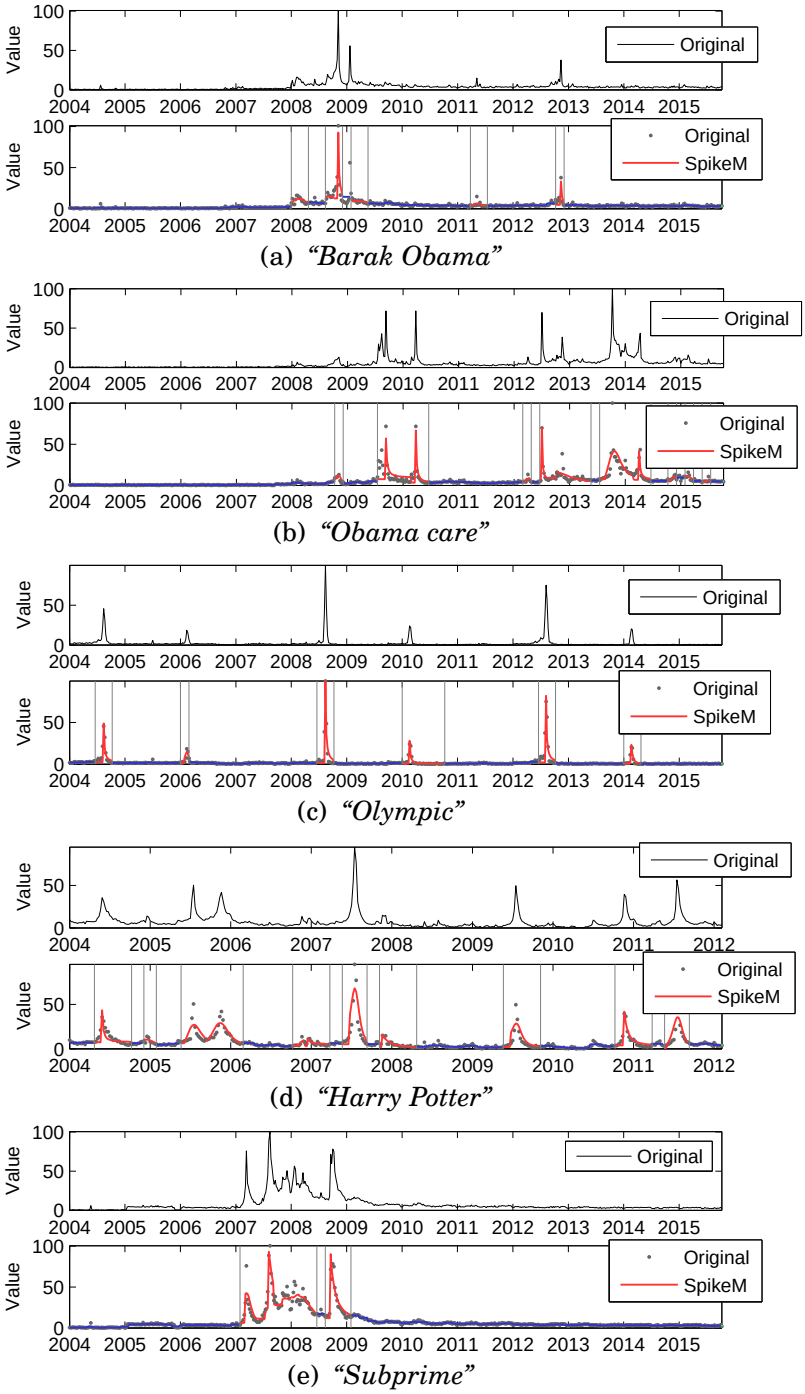
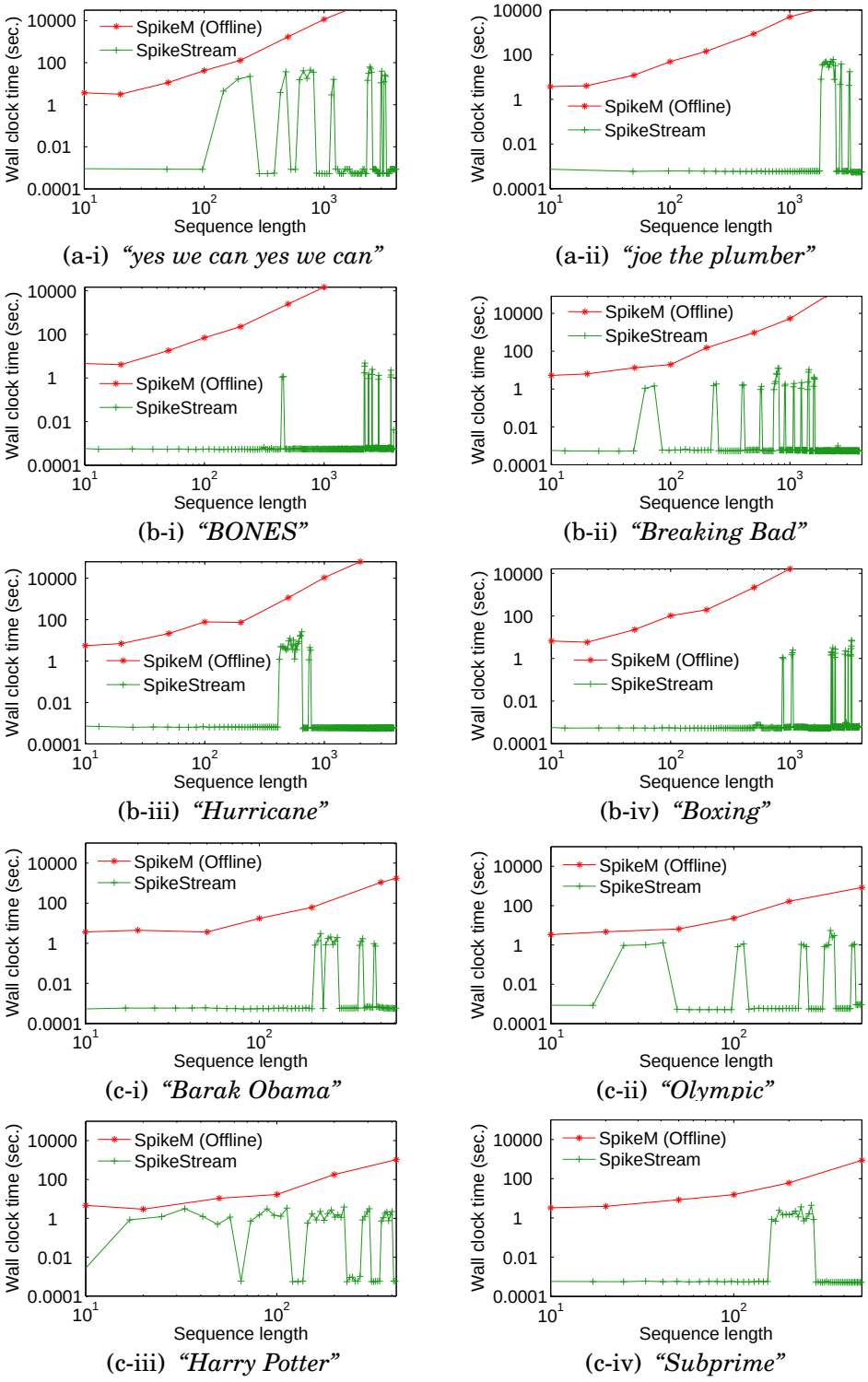


Fig. 18. Real-time monitoring of information diffusion in the *Google* streams. The top figure shows the original event stream, and the bottom figure shows our discoveries. Note that the optimal diffusion segments are shown as red lines.



(a-i) *"yes we can yes we can"* (a-ii) *"joe the plumber"*

(b-i) *"BONES"* (b-ii) *"Breaking Bad"*

(b-iii) *"Hurricane"* (b-iv) *"Boxing"*

(c-i) *"Barak Obama"* (c-ii) *"Olympic"*

(c-iii) *"Harry Potter"* (c-iv) *"Subprime"*

Fig. 19. Scalability of SPIKESTREAM: Wall clock time vs. sequence length n, shown in log-log scales.

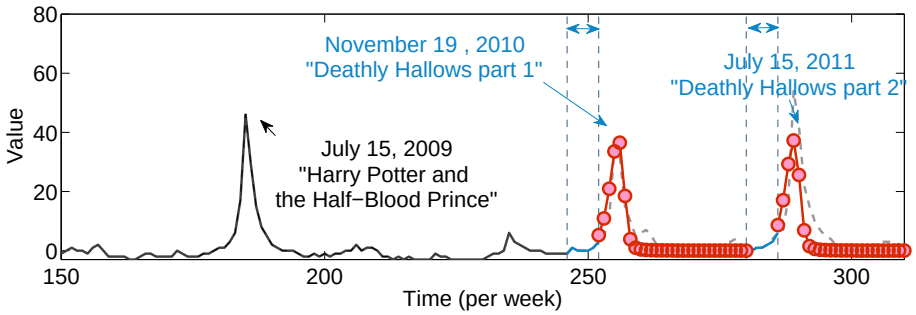


Fig. 20. "What-if" forecasting for the Harry Potter series. We trained parameters by using (a) the first spike around July 15, 2009 ("Harry Potter and the half-blood prince", shown as a black solid line), and (b) access volume of two months before the release (blue lines with double arrows around time $n = 250, 280$), and then forecasted the following two spikes. Here, the forecasted results are shown as red lines, and the original spikes are shown as gray dashed lines.

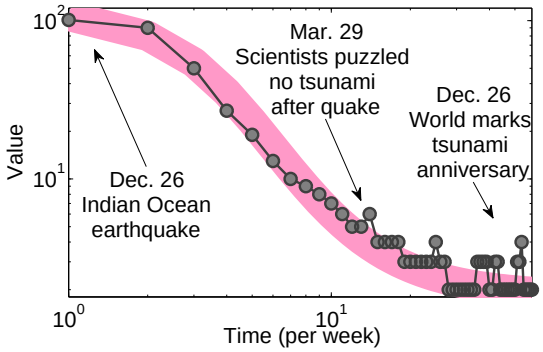


Fig. 21. Outlier detection on *Google* dataset (in log-log scale). Notice that the biggest spike, "world marks tsunami anniversary" occurred after one year (i.e., 52 weeks later).

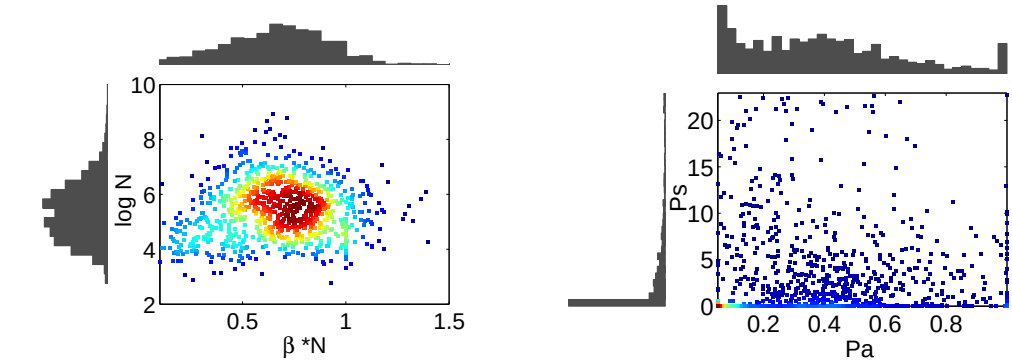
Figure 20 shows that our model successfully captures the two sequel spikes and peak points n_{mode}, especially for around rise parts and peak points, which are the most important for the spike forecasts.

6.2. Outlier detection

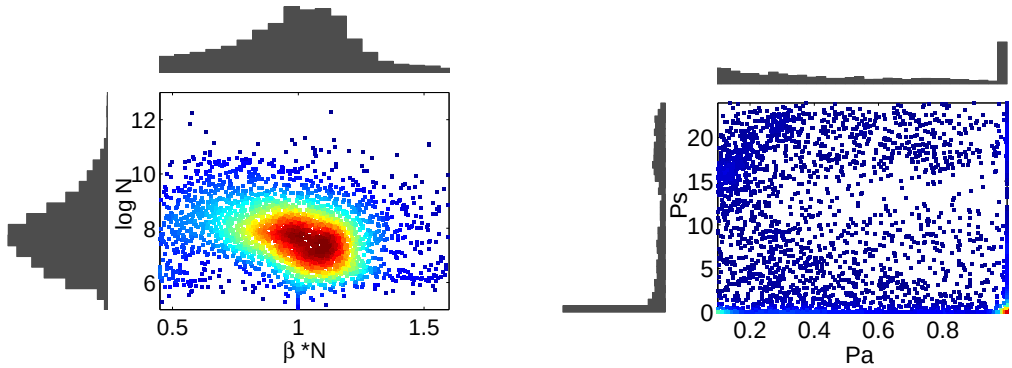
Since SPIKEM has a very high fitting accuracy on real datasets (described in section 5), another natural application would be anomaly detection. Figure 21 shows the fitting result of Figure 11 (a), in a **log-log** scale. Note that the black circles are the original sequence, and the pink line is our model fitting. We can visually observe that there are several points that do not overlap the model. For example, (a) on March 29, there is one spike, since another earthquake occurred on March 28. (b) There is a huge spike on December 26, 2005, which is exactly one year after the Indian Ocean earthquake.

6.3. Reverse engineering

Most importantly, our model can provide an intuitive explanation such as the potential number of interested bloggers, and the quality of news. Figure 22 shows the scatter/pdf



(a-1) *MemeTracker*: $\beta * N$ vs. $\log(N)$ (a-2) *MemeTracker*: P_a vs. P_s



(b-1) *Twitter*: $\beta * N$ vs. $\log(N)$ (b-2) *Twitter*: P_a vs. P_s

Fig. 22. Reverse engineering: scatter/pdf plots of several parameters: (1) $\beta * N$ vs. $\log(N)$ and (2) P_a vs. P_s over 1,000 memes/hashtags. (a) *MemeTracker*: total potential bloggers $N \simeq 1,000$, and strength of the infection $\beta * N \simeq 0.6 - 1.2$. Almost all the memes have clear daily periodicity with high activities around 6pm (i.e., $P_s \simeq 0$). (b) *Twitter*: similar trends except more spread in P_s, possibly, due to multiple time zone. Also see the text for more observations.

plots of several parameters: (1) $\beta * N$ vs. $\log N$, (2) P_a vs. P_s. Here we report our discoveries on *MemeTracker* and *Twitter* datasets (see Figure 22).

OBSERVATION 3 (TOTAL POPULATION OF BLOGGERS). *The total populations of potential bloggers/users N are almost the same for both datasets (around $N = 1,000 - 2,000$).*

We also note that they are skewed to the right, i.e., there is a long tail of larger values.

OBSERVATION 4 (STRENGTH OF FIRST INFECTION). *The strength of the "first burst" is $\beta * N \simeq 0.6 - 1.2$, for each dataset.*

The above two observations agree with the intuition: we can see common behavior for *MemeTracker* and *Twitter*, which means that they have similar characteristics in terms of social activities.

OBSERVATION 5 (COMMON ACTIVITY AND PERIODICITY). *Typical user behavior is to have a daily periodicity with (a) phase shift $P_s = 0$ (small population during early morning, large population at peak point, 6pm) for MemeTracker, while (b) more spread in P_s.*

Note that almost all spikes have a daily periodicity in both datasets. The only the difference between the two datasets is that *Twitter* has several P_s values. This is because *Twitter* has multiple time zones (e.g., US, UK, Australia, and India).

7. RELATED WORK

We provide a survey of the related literature, which falls broadly into three categories: (a) time series analysis, (b) influence propagation and (c) burst detection.

7.1. Time series Analysis

There is a lot of interest in mining time series and data streams [Sakurai et al. 2015; Matsubara and Sakurai 2016; Box et al. 1994; Papadimitriou and Yu 2006; Aggarwal 2014; Jr. et al. 2014; Wang et al. 2006; Ferlez et al. ; Chen and Ng 2004; Papadimitriou et al. 2005; Vlachos et al. 2005; Matsubara et al. 2014a; Sakurai et al. 2005b; Toyoda et al. 2013; Chang et al. 2014; Lee et al. 2007; Sun et al. 2006; Davidson et al. 2013]. Traditional approaches applied to data mining include auto-regression (AR), and variations [Li et al. 2011], linear dynamical systems (LDS), Kalman filters (KF) and their variants [Jain et al. 2004; Li et al. 2009; Li et al. 2010; Tao et al. 2004]. With respect to the non-linear time-series analysis, the work in [Matsubara et al. 2013] uses the power laws and fractal dimensions to characterize the temporal patterns of trajectories, while Korn et al. [Korn et al. 2006] presented a scalable algorithm for the power-law/fractal modeling of data streams.

Non-linear methods for forecasting tend to be hard to interpret, because they rely on nearest-neighbor search [Chakrabarti and Faloutsos 2002], interpolation in state-space [Sauer 1994], or artificial neural networks [Weigend and Gerschenfeld 1994]. Similarity search, indexing and pattern discovery in time sequences have also attracted huge interest [Faloutsos et al. 1994; Kahveci and Singh 2001; Gilbert et al. 2001; Patel et al. 2002; Keogh et al. 2004; Papadimitriou and Yu 2006; Lin et al. 2004; Vlachos et al. 2009; Papapetrou et al. 2011; Sakurai et al. 2007; Sakurai et al. 2005a; Matsubara et al. 2009].

Regarding large-scale time-series mining, TriMine [Matsubara et al. 2012a] is a scalable method for forecasting co-evolving multiple (thousands of) sequences, while, [Matsubara et al. 2014] developed a fully-automatic mining algorithm for co-evolving sequences. Rakthanmanon et al. [Rakthanmanon et al. 2012] proposed a similarity search algorithm for "trillions of time series" under the DTW distance. Yang et al. [Yang et al. 2014] developed a new model for mining time-evolving event sequences. As regards parameter-free mining, the work in [Böhm et al. 2008; Chakrabarti et al. 2004; Tatti and Vreeken 2012], focused on summarization and clustering based on the MDL principle. However, none of these methods specifically focused on modeling bursts.

7.2. Influence propagation

In recent years, there has been an explosion of interest in mining and analyses of blogs, online news, social media, epidemics and online user activities [Sakurai et al. 2016; Leskovec et al. 2007; Beutel et al. 2012; Prakash et al. 2012; Leskovec et al. 2009; Yang and Leskovec 2010; Kumar et al. 2010; Prakash et al. 2011; Kempe et al. 2003a; Tong et al. 2010; Goetz et al. 2009; Koren 2008; Shmueli et al. 2012; Lu et al. 2010; Eirinaki and Vazirgiannis 2003; Cui et al. 2013; Gruhl et al. 2004; Guha et al. 2004;

Weng et al. 2010b; Saez-Trumper et al. 2012], and recently the reverse problem ('find who started it') [Lappas et al. 2010; Shah and Zaman 2011]. The canonical textbook for epidemiological models such as SI and SIR models is Anderson and May [Anderson and May 1991]. The power-law decay of influence has been reported in blogs [McGlohon et al. 2007], with a exponent of -1.5. Barabasi and his colleagues reported exponents of -1 and -1.5, for the response time in correspondence [Barabasi 2005].

Analysis of information diffusion and influence propagation in social networks have also attracted considerable interest [Lou and Tang 2013; Weng et al. 2010a; Kwon et al. 2013; Cha et al. 2010; Kempe et al. 2003b; Chen et al. 2009; Leskovec et al. 2007a]. Yang et al. [Yang and Leskovec 2011] examined patterns of temporal behavior on Twitter, blog posts and news media articles. They did an empirical classification of rise-and-fall patterns, and found six typical patterns that popularity of online content exhibits (see Figure 1). The work in [Matsubara et al. 2012b] studied the rise and fall patterns in the information diffusion process through online social media. The work in [Figueiredo et al. 2014] investigated the effect of revisits on content popularity, while [Ribeiro 2014] focused on the daily number of active users, and studied the mechanisms of the growth and death of membership-based websites. Prakash et al. [Prakash et al. 2012] described a case where two competing products/ideas spreading over the network, and provided a theoretical analysis of the propagation model (winner takes all: WTA) for arbitrary graph topology. FUNNEL [Matsubara et al. 2014b] is a non-linear model for spatially co-evolving epidemic tensors, while EcoWeb [Matsubara et al. 2015] is the first attempt to bridge the theoretical modeling of a biological ecosystem and user activities on the Web. The work in [Figueiredo et al. 2014] investigated the effect of revisits on content popularity, while [Ribeiro 2014] focused on the daily number of active users. For online activity analysis, Gruhl et al. [Gruhl et al. 2005] explored online "chatter" (e.g., blogging) activity, and measured the actual sales ranks on Amazon.com. Ginsberg et al. [Ginsberg et al. 2009] examined a large number of search engine queries tracking influenza epidemics. They reported that the evolutions of search engine keywords are highly correlated with actual flu virus activity. The work reported in [Matsubara et al. 2016; Choi and Varian 2012; Preis et al. 2013; Goel et al. 2010] studied keyword volume, to predict online user behavior.

7.3. Burst detection

Remotely related to our work are the efforts to spot bursts. This includes the work of Kleinberg [Kleinberg 2002], the algorithm of Zhu and Shasha [Zhu and Shasha 2003], and the algorithm of Parikh et al. [Parikh and Sundaresan 2008]. None of the above gives a parsimonious model for describing the activity in a network.

8. CONCLUSIONS

In this paper, we study the rise-and-fall patterns in information diffusion process through online media. We present SPIKEM, a general, accurate and succinct model that explains the rise-and-fall patterns. Our proposed SPIKEM has the following appealing advantages:

— **Unification power**: it includes earlier patterns (K-SC) and models as special cases (i.e., the SI and SIR models), and it can handle an arbitrary graph topology;
— **Practicality**: it matches the behavior of numerous, diverse, real datasets, including the power law decay and much more beyond;
— **Parsimony**: our model requires only a handful of parameters;

— **Usefulness**: we describe how to use our model to do 'short-term' forecasting, to answer what-if scenarios, to spot outliers, and to learn more about the mechanisms of the spikes. We also introduce SPIKESTREAM, which identifies all the important information-diffusion spikes in a large collection of event stream.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments on our original submission. We also thank Jaewon Yang and Jure Leskovec for providing the details of the six clusters in Figure 1. This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research Number JP15H02705, JP16K12430, JP26280112, JP26730060, PRESTO JST, and the MIC/SCOPE #162110003. This material is based upon work supported by the Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053, and the National Science Foundation under Grant No. IIS-1017415. This paper is based on work partially supported by the National Science Foundation (IIS-1353346), the National Endowment for the Humanities (HG-229283-15), ORNL (Task Order 4000143330) and from the Maryland Procurement Office (H98230-14-C- 0127), and a Facebook faculty gift.

REFERENCES

Charu C. Aggarwal. 2014. The setwise stream classification problem. In *KDD*. 432–441.

Roy M. Anderson and Robert M. May. 1991. *Infectious Diseases of Humans*. Oxford University Press.

Albert L. Barabasi. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435 (2005). http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0505371

Frank M. Bass. 1969. A New Product Growth for Model Consumer Durables. *Management Science* 15, 5 (1969), 215–227.

Alex Beutel, B. Aditya Prakash, Roni Rosenfeld, and Christos Faloutsos. 2012. Interacting viruses in networks: can both survive?. In *KDD*. 426–434.

Christian Böhm, Christos Faloutsos, and Claudia Plant. 2008. Outlier-robust clustering using independent components. In *SIGMOD*. 185–198.

George E.P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. 1994. *Time Series Analysis: Forecasting and Control* (3rd ed.). Prentice Hall, Englewood Cliffs, NJ.

F. Brauer and C. Castillo-Chavez. 2001. *Mathematical models in population biology and epidemiology*. Vol. 40. Springer Verlag, New York.

Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and P. Krishna Gummadi. 2010. Measuring User Influence in Twitter: The Million Follower Fallacy. In *ICWSM*.

Deepay Chakrabarti and Christos Faloutsos. 2002. F4: Large-Scale Automated Forecasting using Fractals. *CIKM* (2002).

Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. 2004. Fully automatic cross-associations. In *KDD*. 79–88.

Yi Chang, Makoto Yamada, Antonio Ortega, and Yan Liu. 2014. Ups and Downs in Buzzes: Life Cycle Modeling for Temporal Pattern Discovery. In *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*. 749–754.

Lei Chen and Raymond T. Ng. 2004. On The Marriage of Lp-norms and Edit Distance. In *VLDB*. 792–803.

Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*. 199–208.

Hyunyoung Choi and Hal Ronald Varian. 2012. Predicting the Present with Google Trends. *The Economic Record* 88, s1 (2012), 2–9.

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

Peng Cui, Shifei Jin, Linyun Yu, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2013. Cascading outbreak prediction in networks: a data-driven approach. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 901–909.

Ian N. Davidson, Sean Gilpin, Owen T. Carmichael, and Peter B. Walker. 2013. Network discovery via constrained tensor analysis of fMRI data. In *KDD*. 194–202.

Magdalini Eirinaki and Michalis Vazirgiannis. 2003. Web mining for web personalization. *ACM Trans. Internet Techn.* 3, 1 (2003), 1–27.

Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast Subsequence Matching in Time-Series Databases. In *SIGMOD*. 419–429.

Jure Ferlez, Christos Faloutsos, Jure Leskovec, Dunja Mladenic, and Marko Grobelnik. Monitoring Network Evolution using MDL. In *ICDE*. 1328–1330.

Flavio Figueiredo, Jussara M. Almeida, Yasuko Matsubara, Bruno Ribeiro, and Christos Faloutsos. 2014. Revisit Behavior in Social Media: The Phoenix-R Model and Discoveries. In *PKDD*. 386–401.

Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. 2001. Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries. In *VLDB*. 79–88.

Jeremy Ginsberg, Matthew Mohebbi, Rajan Patel, Lynnette Brammer, Mark Smolinski, and Larry Brilliant. 2009. Detecting influenza epidemics using search engine query data. *Nature* 457 (2009), 1012–1014.

Sharad Goel, Jake Hofman, Sebastien Lahaie, David Pennock, and Duncan Watts. 2010. Predicting Consumer Behavior with Web Search. *PNAS* (2010).

Michaela Goetz, Jure Leskovec, Mary McGlohon, and Christos Faloutsos. 2009. Modeling Blog Dynamics. In *ICWSM*.

Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2005. The Predictive Power of Online Chatter. In *KDD*. 78–87.

D. Gruhl, David Liben-Nowell, R. Guha, and A. Tomkins. 2004. Information diffusion through blogspace. *SIGKDD Explor. Newsl.* 6, 2 (December 2004), 43–52. DOI:http://dx.doi.org/10.1145/1046456.1046462

R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *WWW*. 403–412.

A. G. Hawkes and D. Oakes. 1974. A cluster representation of a self-exciting process. *J. Appl. Prob.* 11 (1974), 493–503.

Herbert W. Hethcote. 2000. The Mathematics of Infectious Diseases. *SIAM Rev.* 42, 4 (Dec. 2000), 599–653.

E.A. Jackson. 1992. *Perspectives of Nonlinear Dynamics:*. Cambridge University Press.

Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. 2004. Adaptive stream resource management using Kalman Filters. In *SIGMOD*. 11–22. DOI:http://dx.doi.org/10.1145/1007568.1007573

Roberto Lourenco Jr., Adriano Veloso, Adriano M. Pereira, Wagner Meira Jr., Renato Ferreira, and Srinivasan Parthasarathy. 2014. Economically-efficient sentiment stream analysis. In *SIGIR*. 637–646.

Tamer Kahveci and Ambuj K. Singh. 2001. An Efficient Index Structure for String Databases. In *Proceedings of VLDB*. 351–360.

D. Kempe, J. Kleinberg, and E. Tardos. 2003a. Maximizing the Spread of Influence through a Social Network. In *KDD*.

David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003b. Maximizing the spread of influence through a social network. In *SIGKDD*. 137–146.

Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani. 2001. An Online Algorithm for Segmenting Time Series. In *ICDM*. 289–296.

Eamonn J. Keogh, Themis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. 2004. Indexing Large Human-Motion Databases. In *VLDB*. 780–791.

Jon M. Kleinberg. 2002. Bursty and hierarchical structure in streams. In *KDD*. 91–101.

Steven Klepper. 1996. Entry, Exit, Growth, and Innovation over the Product Life Cycle. *American Economic Review* 86, 3 (June 1996), 562–83.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*. 426–434.

Flip Korn, S. Muthukrishnan, and Yihua Wu. 2006. Modeling skew in data streams. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, New York, NY, USA, 181–192.

Ravi Kumar, Mohammad Mahdian, and Mary McGlohon. 2010. Dynamics of conversations. In *SIGKDD*. 553–562.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent Features of Rumor Propagation in Online Social Media. In *ICDM*. 1103–1108.

Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. 2010. Finding effectors in social networks. In *KDD*. 1059–1068.

Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. 2007. Trajectory clustering: a partition-and-group framework. In *SIGMOD*. 593–604.

W. Leontief. 1986. *Input-output economics*. Oxford University Press.

Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *TWEB* 1, 1 (2007).

Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *KDD*. 497–506.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker Graphs: An Approach to Modeling Networks. *J. Mach. Learn. Res.* 11 (March 2010), 985–1042.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. 2007a. Cost-effective outbreak detection in networks. In *SIGKDD*. 420–429.

Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S. Glance, and Matthew Hurst. 2007b. Patterns of Cascading Behavior in Large Blog Graphs. In *SDM*.

K. Levenberg. 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics* II, 2 (1944), 164–168.

Lei Li, Chieh-Jan Mike Liang, Jie Liu, Suman Nath, Andreas Terzis, and Christos Faloutsos. 2011. ThermoCast: A Cyber-Physical Forecasting Model for Data Centers. In *KDD*.

Lei Li, James McCann, Nancy Pollard, and Christos Faloutsos. 2009. DynaMMo: Mining and Summarization of Coevolving Sequences with Missing Values. In *KDD*.

Lei Li, B. Aditya Prakash, and Christos Faloutsos. 2010. Parsimonious Linear Fingerprinting for Time Series. *PVLDB* 3, 1 (2010), 385–396.

Jessica Lin, Eamonn J. Keogh, Stefano Lonardi, Jeffrey P. Lankford, and Donna M. Nystrom. 2004. Visually mining and monitoring massive time series. In *KDD*. 460–469.

Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. *J. Amer. Statist. Assoc.* 106, 496 (2011), 1513–1527.

Tiancheng Lou and Jie Tang. 2013. Mining structural hole spanners through information diffusion in social networks. In *WWW*. 825–836.

Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. 2010. Exploiting social context for review quality prediction. In *WWW*. 691–700.

Yasuko Matsubara, Lei Li, Evangelos E. Papalexakis, David Lo, Yasushi Sakurai, and Christos Faloutsos. 2013. F-Trail: Finding Patterns in Taxi Trajectories. In *PAKDD*. 86–98.

Yasuko Matsubara and Yasushi Sakurai. 2016. Regime Shifts in Streams: Real-time Forecasting of Coevolving Time Sequences. In *KDD*. 1045–1054.

Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. 2014. AutoPlait: Automatic Mining of Coevolving Time Sequences. In *SIGMOD*.

Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. 2015. The Web as a Jungle: Non-Linear Dynamical Systems for Co-evolving Online Activities. In *WWW*.

Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. 2016. Non-Linear Mining of Competing Local Activities. In *WWW*.

Yasuko Matsubara, Yasushi Sakurai, Christos Faloutsos, Tomoharu Iwata, and Masatoshi Yoshikawa. 2012a. Fast mining and forecasting of complex time-stamped events. In *KDD*. 271–279.

Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. 2012b. Rise and fall patterns of information diffusion: model and implications. In *KDD*. 6–14.

Yasuko Matsubara, Yasushi Sakurai, Naonori Ueda, and Masatoshi Yoshikawa. 2014a. Fast and Exact Monitoring of Co-Evolving Data Streams. In *ICDM 2014*. 390–399.

Yasuko Matsubara, Yasushi Sakurai, Willem G. van Panhuis, and Christos Faloutsos. 2014b. FUNNEL: automatic mining of spatially coevolving epidemics. In *KDD*. 105–114.

Yasuko Matsubara, Yasushi Sakurai, and Masatoshi Yoshikawa. 2009. Scalable Algorithms for Distribution Search. In *ICDM*. 347–356.

Julian J. McAuley and Jure Leskovec. 2012. Learning to Discover Social Circles in Ego Networks. In *NIPS*. 548–556.

Mary McGlohon, Jure Leskovec, Christos Faloutsos, Matthew Hurst, and Natalie Glance. 2007. Finding Patterns in Blog Shapes and Blog Evolution. In *International Conference on Weblogs and Social Media*. Boulder, Colo.

M.A. Nowak. 2006. *Evolutionary Dynamics*. Harvard University Press.

Spiros Papadimitriou, Anthony Brockwell, and Christos Faloutsos. 2003. Adaptive, Hands-Off Stream Mining. In *VLDB*. 560–571.

Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. 2005. Streaming Pattern Discovery in Multiple Time-Series. In *VLDB*. 697–708.

Spiros Papadimitriou and Philip S. Yu. 2006. Optimal multi-scale patterns in time series streams. In *SIGMOD*. 647–658.

Evangelos E. Papalexakis, Tudor Dumitras, Duen Horng (Polo) Chau, B. Aditya Prakash, and Christos Faloutsos. 2013. Spatio-temporal mining of software adoption & penetration. In *ASONAM*. 878–885.

Panagiotis Papapetrou, Vassilis Athitsos, Michalis Potamias, George Kollios, and Dimitrios Gunopulos. 2011. Embedding-based subsequence matching in time-series databases. *ACM Trans. Database Syst.* 36, 3 (2011), 17.

Nish Parikh and Neel Sundaresan. 2008. Scalable and near real-time burst detection from eCommerce queries. In *KDD*. 972–980.

Pranav Patel, Eamonn J. Keogh, Jessica Lin, and Stefano Lonardi. 2002. Mining Motifs in Massive Time Series Databases. In *Proceedings of ICDM*. 370–377.

B. Aditya Prakash, Alex Beutel, Roni Rosenfeld, and Christos Faloutsos. 2012. Winner takes all: competing viruses or ideas on fair-play networks. In *WWW*. 1037–1046.

B. Aditya Prakash, Deepayan Chakrabarti, Michalis Faloutsos, Nicholas Valler, and Christos Faloutsos. 2011. Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks. In *ICDM*.

B. Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. 2012. Spotting Culprits in Epidemics: How Many and Which Ones?. In *ICDM*. 11–20.

Tobias Preis, Helen Susannah Moat, and H. Eugene Stanley. 2013. Quantifying Trading Behavior in Financial Markets Using Google Trends. *Sci. Rep.* 3 (04 2013).

Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo E. A. P. A. Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn J. Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*. 262–270.

Bruno Ribeiro. 2014. Modeling and predicting the growth and death of membership-based websites. In *WWW*. 653–664.

Diego Saez-Trumper, Giovanni Comarela, Virgílio Almeida, Ricardo Baeza-Yates, and Fabrício Benevenuto. 2012. Finding Trendsetters in Information Networks. In *KDD*. ACM, 1014–1022.

Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. 2007. Stream Monitoring under the Time Warping Distance. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, April 15-20, 2007, The Marmara Hotel, Istanbul, Turkey*. 1046–1055.

Yasushi Sakurai, Yasuko Matsubara, and Christos Faloutsos. 2015. Mining and Forecasting of Big Time-series Data. In *SIGMOD, Tutorial*. 919–922.

Yasushi Sakurai, Yasuko Matsubara, and Christos Faloutsos. 2016. Mining Big Time-series Data on the Web. In *WWW, Tutorial*. 1029–1032.

Yasushi Sakurai, Spiros Papadimitriou, and Christos Faloutsos. 2005a. BRAID: Stream Mining through Group Lag Correlations. In *SIGMOD Conference*. Baltimore, MD, USA, 599–610.

Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. 2005b. FTW: Fast Similarity Search under the Time Warping Distance. In *PODS*. Baltimore, Maryland, 326–337.

T. Sauer. 1994. Time series prediction using delay coordinate embedding. In *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld (Eds.). Addison-Wesley.

Devavrat Shah and Tauhid Zaman. 2011. Rumors in a Network: Who's the Culprit? *IEEE Transactions on Information Theory* 57, 8 (2011), 5163–5181.

Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. 2012. Care to comment?: recommendations for commenting on news stories. In *WWW*. 429–438.

Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: dynamic tensor analysis. In *KDD*. 374–383.

Yufei Tao, Christos Faloutsos, Dimitris Papadias, and Bin Liu. 2004. Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In *SIGMOD*. 611–622.

Nikolaj Tatti and Jilles Vreeken. 2012. The long and the short of it: summarising event sequences with serial episodes. In *KDD*. 462–470.

Hanghang Tong, B. Aditya Prakash, Charalampos E. Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. 2010. On the Vulnerability of Large Graphs. In *ICDM*.

Machiko Toyoda, Yasushi Sakurai, and Yoshiharu Ishikawa. 2013. Pattern discovery in data streams under the time warping distance. *VLDB J.* 22, 3 (2013), 295–318.

Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2005. Elastic Translation Invariant Matching of Trajectories. *Mach. Learn.* 58, 2-3 (Feb. 2005), 301–334. DOI:http://dx.doi.org/10.1007/s10994-005-5830-9

Michail Vlachos, Suleyman Serdar Kozat, and Philip S. Yu. 2009. Optimal Distance Bounds on Time-Series Data. In *SDM*. 109–120.

Haixun Wang, Jian Yin, Jian Pei, Philip S. Yu, and Jeffrey Xu Yu. 2006. Suppressing model overfitting in mining concept-drifting data streams. In *KDD*. 736–741.

Peng Wang, Haixun Wang, and Wei Wang. 2011. Finding semantics in time series. In *SIGMOD Conference*. 385–396.

Andreas S. Weigend and Neil A. Gerschenfeld. 1994. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison Wesley.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010a. TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*. 261–270.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010b. TwitterRank: Finding Topic-sensitive Influential Twitterers. In *WSDM*. ACM, 261–270.

Jaewon Yang and Jure Leskovec. 2010. Modeling Information Diffusion in Implicit Networks. In *ICDM*. 599–608.

Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *WSDM*. 177–186.

Jaewon Yang, Julian J. McAuley, Jure Leskovec, Paea LePendu, and Nigam Shah. 2014. Finding progression stages in time-evolving event sequences. In *WWW*. 783–794.

Yunyue Zhu and Dennis Shasha. 2003. Efficient elastic burst detection in data streams. In *KDD*. 336–345.

Received February 2007; revised March 2009; accepted June 2009