2015 IEEE International Conference on Data Mining

Modeling Adoption and Usage of Competing Products

Isabel Valera
MPI for Software Systems
ivalera@mpi-sws.org

Manuel Gomez-Rodriguez
MPI for Software Systems
manuelgr@mpi-sws.org

***Abstract*—The emergence and wide-spread use of online social networks has led to a dramatic increase on the availability of social activity data. Importantly, this data can be exploited to investigate, at a microscopic level, some of the problems that have captured the attention of economists, marketers and sociologists for decades, such as, *e.g.*, product adoption, usage and competition.**

In this paper, we propose a continuous-time probabilistic model, based on temporal point processes, for the adoption and frequency of use of competing products, where the frequency of use of one product can be modulated by those of others. This model allows us to efficiently simulate the adoption and recurrent usages of competing products, and generate traces in which we can easily recognize the effect of social influence, recency and competition. We then develop an inference method to efficiently fit the model parameters by solving a convex program. The problem decouples into a collection of smaller subproblems, thus scaling easily to networks with hundred of thousands of nodes. We validate our model over synthetic and real diffusion data gathered from Twitter, and show that the proposed model does not only provides a good fit to the data and more accurate predictions than alternatives but also provides interpretable model parameters, which allow us to gain insights into some of the factors driving product adoption and frequency of use.

I. Introduction

There is a long history of work in economics and marketing on studying product adoption, product competition and product life cycle [1]. This work has typically developed mathematical models, such as, the Bass diffusion model [2], the probit model [3], or models based on information cascades [4], that attempt to capture the macroscopic evolution of a product or set of competing products but not its (their) microscopic evolution, *i.e.*, product adoptions and recurrent usage by specific individuals. This has been in part due to the lack of large-scale fine-grained product adoption and product usage data, which would allow to propose and validate microscopic models at scale.

The emergence and wide-spread use of online social networks has led to a dramatic increase on the availability of large-scale fine-grained social activity data, in which all individual adoptions and subsequent uses of a product are often *observable*. This availability has opened up a great opportunity for a paradigm shift, where we investigate both the macroscopic and microscopic dynamics of product adoption and usage. As a consequence, there recent empirical studies have investigated the adoption and usage of competing products [5] in online social networks. Therefore, it is necessary to develop microscopic models of product adoption and usage, which are largely non-existent to date.

A. Our approach

We introduce a continuous-time probabilistic model, based on temporal point processes, for the adoption and frequency of use of competing products in online social networks. The model captures several intuitive key factors, which has been previously discussed in the literature:

I. **Social Influence.** Whenever a user observes that one or more of her neighbors are using a product, she may decide to start using the product [6].
II. **Recency.** If a user has used a product in the past, she is more likely to use the product in the future than if she has never used it [7].
III. **Competition.** If a user who is currently using a product p_1 observes that one or more of her neighbors use a competing product p_2, she may decide to switch from product p_1 to p_2 [8].

To simultaneously capture the above mentioned factors, which have been typically studied separately, our model leverages multivariate Hawkes processes [9]. Although Hawkes processes have been used for modeling positive social influence before [10], [11], [12], [13], [14], the key innovation here is that we explicitly model recency and competition, and allow for both positive and negative social influence. As a consequence, our design is especially well fitted to model adoption and recurrent usage of competing products.

Perhaps surprisingly, the flexibility of our model of competing products does not prevent us from efficiently simulate from the model, and learn the model parameters from real world product fine-grained usage data:

- **Efficient simulation.** We develop an efficient sampling method that leverages Ogata's method [15], and scales linearly in the number of products. Our method exploits the sparsity of social networks to reduce the overall complexity (refer to Algorithm 1).
- **Efficient parameter estimation.** We fit the parameters of the model using historical data by solving a maximum likelihood estimation problem, which reduces to solving a convex program. Importantly, the problem decouples in several smaller problems, allowing for natural parallelization so that we can fit the parameters of the model for networks with hundred of thousands of nodes.

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.40

Moreover, we validate our model using both synthetic and real social activity data, and show that:

1) Our model can generate product use events that obey temporal patterns described in related literature [16], [17], such as temporal bursts of events or product switching.
2) Our regularized maximum likelihood estimation method can recover the parameters of the model with an accuracy that increases as we record more product uses over time.
3) Our model provides a significantly better fit and more accurate product use event predictions in real data gathered from Twitter [18] than several alternatives [7], [16]. In particular, we experiment with one type of competing product, url shortening services [5], and one type of conventions, the way Twitter users indicated a tweet was a retweet back in 2009 [19].
4) Our model allows us to gain insights into some of the factors driving product adoption and frequency of use. For example, we find that the usage of popular products is typically driven by recency, however, an exposure to a less popular product can have a strong inhibiting effect on future uses of popular products.

B. *Related work*

Several models of adoption of competing products in online social networks have been proposed in the computer science literature in recent years [8], [20], [21]. In particular, Bharathi et al. [8] extend the independent cascade model, Dubey et al. [20] propose a local quasilinear model, and Goyal et al. [21] consider a broad class of local influence processes. Moreover, competition has been also studied in other contexts: meme diffusion [22], rumor propagation [23], spread of misinformation [24], opinion formation [25], [26] and disease spread [27], [28]. However, most previous work share the following limitations, which we tackle in our work. First, they consider discrete-time sequential models of competition, which are difficult to estimate accurately from real world data [29], [30], [12]. Instead, we model competition using a continuous-time asynchronous model, designed to naturally fit the event data we record (*i.e.*, the times in which users use products). Second, they consider each user to adopt only one *contagion* (be it a product, a meme, rumor, opinion or disease), the one she adopted *first*, and *use* it only once. In contrast, we allow users to use it once or several times and adopt different competing contagions simultaneously. That is, we model users' frequency of use, and how this *frequency* depends on what users *observe* over time. Hence, we are able to model non-progressive phenomena [6]. Third, they only allow for pairwise interactions between contagions, being able to study competition between only two different products. In contrast, we consider n-ary interactions, which consider the interactions/competition among all the competing products in the network.

Repeat online behavior have been studied in the specific contexts of repeat queries in web search [31], [32] or repeat website visits [33], [34]. Only very recently, Anderson et al. [7] propose a fairly general discrete-time model of online reconsumption of goods, such as songs or movies, and experiences. However, this line of work does not model social influence nor competition and typically focus on goods, experiences or behaviors a user may get satiated or bored of, rather than products a user may repeatedly keep using for long periods of time.

Last, Hawkes processes have been also applied to a large variety of problems in which self-excitement plays a fundamental role: earth quake prediction [35], crime prediction [36], computational neuroscience [37], or high frequency trading [38]. However, to the best of our knowledge, Hawkes processes have not been applied in the context of adoption and usage of competing products.

II. Proposed model

In this section, we formulate our probabilistic model of adoption and frequency of use of competing products. We start by revisiting the foundations of temporal point processes, and then describe the specifics of our model.

A. *Temporal Point Processes*

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. Many different types of data produced in online social networks and the Web can be represented as temporal point processes, *e.g.*, the times when a user uses a product. A temporal point process can be equivalently represented as a counting process, $N(t)$, which records the number of events before time t, *e.g.*, the number of product use events before time t. Let the history $\mathcal{H}(t)$ be the list of times of events $\{t_1, t_2, \ldots, t_n\}$ up to but not including time t. Then, in a small time window dt between $[0, t)$, the number of observed event is

$$dN(t) = \sum_{t_i \in \mathcal{H}(t)} \delta(t - t_i)\, dt, \tag{1}$$

where $\delta(t)$ is a Dirac delta function, and hence $N(t) = \int_0^t dN(s)$. It is often assumed that only one event can happen in a small window of size dt, and hence $dN(t) \in \{0, 1\}$.

An important way to characterize temporal point processes is via the conditional intensity function — the stochastic model for the time of the next event given all the times of previous events. The conditional intensity function $\lambda^*(t)$ (intensity, for short) is the conditional probability of observing an event in a small window $[t, t + dt)$ given the history $\mathcal{H}(t)$, *i.e.*,

$$\lambda^*(t)dt = \mathbb{P}\left\{\text{event in } [t, t + dt) | \mathcal{H}(t)\right\}, \tag{2}$$

where * means that the function $\lambda^*(t)$ may depend on the history $\mathcal{H}(t)$. Based on the intensity, one can obtain the conditional expectation of the number of events in the windows $[t, t + dt)$ and $[0, t)$, respectively, as

$$\mathbb{E}[dN(t)|\mathcal{H}(t)] = \lambda^*(t)\, dt, \text{ and} \tag{3}$$

$$\mathbb{E}[N(t)|\mathcal{H}(t)] = \int_0^t \lambda^*(\tau)\, d\tau \tag{4}$$

The functional form of the intensity $\lambda^*(t)$ is often designed to capture the phenomena of interests [39]. For example, the following forms have been argued for in the information propagation and social influence literature [40], [11], [30], [41]:

- **Poisson process.** The intensity is assumed to be independent of the history $\mathcal{H}(t)$ and constant, *i.e.*,
$$\lambda^*(t) = \lambda, \tag{5}$$
where $\lambda \geqslant 0$ is a scale parameter.
- **Weibull renewal process.** The intensity is assumed to be dependent only on the last event $t_n \in \mathcal{H}(t)$ before t, *i.e.*,
$$\lambda^*(t) = k(t - t_n)^{k-1}\lambda^k \geqslant 0, \tag{6}$$
where $k > 0$ is a shape parameter and $\lambda > 0$ is a scale parameter. Remarkably, if the shape parameter is less than 1, then the inter-event time distribution is heavy-tailed, as observed in real social activity data [16].
- **Hawkes Process.** The intensity captures a mutual excitation phenomena between events, *i.e.*,
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} g_\omega(t - t_i), \tag{7}$$
where $g_\omega(t)$ is a nonnegative triggering kernel such that $g_\omega(t) = 0$ for $t < 0$, $\mu \geqslant 0$ is a baseline intensity independent of the history, and the summation of kernel terms is history dependent and a stochastic process by itself. Here, the occurrence of each historical event increases the intensity by a certain amount $\alpha \geqslant 0$.

Now, given a time $t' \geqslant t$, we can also characterize the conditional probability that no event happens during $[t, t')$ and the conditional probability density that an event occurs at time t' using the intensity $\lambda^*(t)$ by means of the following well-known relationships [39]:

$$S^*(t') = \exp\left(-\int_t^{t'} \lambda^*(\tau)\, d\tau\right), \quad \text{and} \tag{8}$$
$$f^*(t') = \lambda^*(t')\, S^*(t'). \tag{9}$$

With these two quantities, we can express the log-likelihood of a list of event times $\{t_1, t_2, \ldots, t_n\}$ in an observation window $[0, T)$ with $T \geqslant t_n$ as

$$\mathfrak{L} = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau, \tag{10}$$

which will be useful for learning the parameters of our model from observed data. With the above background in temporal point processes, we can now proceed to detail the formulation of the proposed model for product adoption and frequency of use, including our choice of specific functional form for the intensity $\lambda^*(t)$.

B. Model for Product Adoption

Given a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of (related) competing products $\mathcal{P}$, we model the generation of product use events by users in the network using temporal point processes. Here, we record each event as a triple

$$e \;:=\; (\,\underset{\substack{\uparrow\\ \text{user}}}{u},\; \overset{\substack{\text{product}\\ \downarrow}}{p},\; \underset{\substack{\uparrow\\ \text{time}}}{t}\,), \tag{11}$$

where the triplet means that the user $u \in \mathcal{V}$ used product $p \in \mathcal{P}$ at time t. We assume that we can observe the triple e for every product use and there are not attribution problems. Given a list of use events $\{e_1 = (u_1, p_1, t_1), \ldots, e_n = (u_n, p_n, t_n)\}$ up to time t, the history $\mathcal{H}_u(t)$ of product use events by user u is

$$\mathcal{H}_u(t) = \{e_i = (u_i, p_i, t_i) | u_i = u \text{ and } t_i < t\}, \tag{12}$$

and the history $\mathcal{H}_{up}(t)$ of use events of product p by user u is

$$\mathcal{H}_{up}(t) = \{e_i = (u_i, p_i, t_i) | u_i = u, p_i = p \text{ and } t_i < t\}. \tag{13}$$

Finally, the entire history of product use events is denoted as $\mathcal{H}(t) := \cup_{u \in \mathcal{V}} \mathcal{H}_u(t)$. Note that each node u does not necessarily use all products $\mathcal{P}$ and can use a specific product p once or several times.

Now, we use a set of counting processes, one for each user and product, to record the generated product use events. More specifically, we denote the set of counting processes as a matrix $\boldsymbol{N}(t)$ of size $|\mathcal{G}| \times |\mathcal{P}|$ for each fixed time point t. The (u, p)-th entry in the matrix, $N_{up}(t) \in \{0\} \cup \mathbb{Z}^+$, counts the number of times user u used product p up to time t. That is, $N_{up}(t) = |\mathcal{H}_{up}(t)|$ is the size of the history $\mathcal{H}_{up}(t)$. Then, we can characterize the users' product usage over time using their corresponding intensities as

$$\mathbb{E}[d\boldsymbol{N}(t) \,|\, \mathcal{H}(t)] = \boldsymbol{\Lambda}^*(t)\, dt, \tag{14}$$

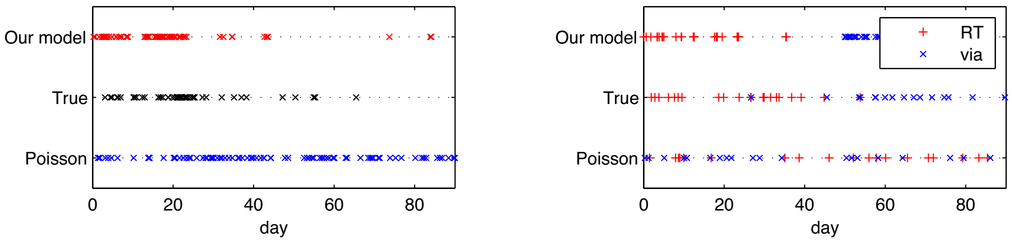
where the matrix $d\boldsymbol{N}(t) := (\ dN_{up}(t)\ )_{u \in \mathcal{V}, p \in \mathcal{P}}$ contains the number of product use events per user and product in the window $[t, t + dt)$, and the matrix $\boldsymbol{\Lambda}^*(t) := (\ \lambda^*_{up}(t)\ )_{u \in \mathcal{V}, p \in \mathcal{P}}$ contains the intensities associated to all the user-product pairs at time t. The sign * means that the intensity matrix $\Lambda^*(t)$ depends on the history $\mathcal{H}(t)$ and each particular intensity $\lambda^*_{up}(t)$ depends on the history node u is exposed to, $\cup_{v \in \mathcal{N}(u) \cup \{u\}} \mathcal{H}_v(t)$, where $\mathcal{N}(u) = \{v \in \mathcal{V} | (v, u) \in \mathcal{E}\}$ denotes the nodes u follows (*i.e.*, her neighbors). In the next section, we specify the functional form of the intensities that accounts for both recency and social influence.

C. Intensity for Product Adoption

To model the intensities, $\Lambda^*(t)$, for product use events, we leverage the multivariate version [9] of the Hawkes process defined by Eq. 7. However, in our work, we modify the original definition of Hawkes processes to allow for both excitation and inhibition phenomena between product use events [42]. In particular, each intensity $\lambda^*_{up}(t)$ takes the following form:

$$\begin{aligned} \lambda^*_{up}(t) := \underbrace{\mu_p}_{\text{spontaneous use}} &+ \underbrace{\sum_{l=1}^{|\mathcal{P}|} a_{lp} \sum_{e_i \in \mathcal{H}_{ul}(t)} g_\omega(t - t_i)}_{\text{recency}} \\ &+ \underbrace{\sum_{l=1}^{|\mathcal{P}|} b_{lp} \sum_{v \in \mathcal{N}(u)} \sum_{e_i \in \mathcal{H}_{vl}(t)} g_\omega(t - t_i)}_{\text{social influence}}, \end{aligned} \tag{15}$$

The first term $\mu_p \geqslant 0$ denotes the base event intensity, which models the spontaneous adoption of a product by a user on her own initiative. In other words, it models adoptions due to other factors different than recency or social influence. The second term models recency, where $a_{lp} \in \mathbb{R}$ denotes the influence that a previous use of a product l by the user has on her intensity for product p. The third term models the



(a) Burstiness

(b) Product switch

Fig. 1. Example of event burstiness and product competition. Our model is able to generate bursts and product switches due to considering mutual excitation and inhibition between events in the intensities, given by Eq. 15, while Poisson processes fail to do so. Since product uses are realizations of a (stochastic) temporal point process, one cannot expect to precisely match the timing of each individual event.

propagation of peer influence over the network. Here, $b_{lp} \in \mathbb{R}$ denotes the influence that a previous use of a product l by one of her neighbors has on the user's intensity for product p. Intuitively, one may expect that $a_{pp}, b_{pp} \geqslant 0$, since previous uses of a product may increase the probability that a user uses the same product in the future, while $a_{lp}, b_{lp} \leqslant 0$, $l \neq p$, since previous uses of a product may decrease the probability that a user use a competing product in the future. For conciseness, we stack all the model parameters in a vector and two matrices, *i.e.*, $\boldsymbol{\mu} := (\mu)_{p\in\mathcal{P}}$, $\boldsymbol{A} = (a_{lp})_{l,p\in\mathcal{P}}$ and $\boldsymbol{B} = (b_{lp})_{l,p\in\mathcal{P}}$. Finally, the triggering kernel $g_\omega(t)$ models the decay of influence over time. Here, for simplicity, we opt for an exponential kernel $g_\omega(t) := \exp(-\omega t)\mathbb{I}[t \geqslant 0]$. However, since our inference method does not depend on this particular choice, one may decide to use more complicated kernels with some desirable characteristics [14] or perform a non-parametric estimation of the kernels [43]. Remarkably, using exponential kernels allow for slightly more efficient likelihood evaluations and, as a result, more efficient model simulation and estimation, as discussed in Sections III and IV.

III. Model Simulation

In this section, we first describe an efficient sampling method for our model of product adoption and then elaborate on the temporal patterns described in related literature [17] that our model is able to reproduce.

A. Simulation method

We simulate samples (product usages) from our model by leveraging Ogata's thinning algorithm [15]. However, an off-the-shelf implementation of Ogata's algorithm would need $O(n^2|\mathcal{V}|^2)$ operations to draw n samples. Here, we first exploit the sparsity of social networks: whenever we sample a new product use event from the model, only a small number of intensity functions, in the local neighborhood of the node that used the product, changes. As a consequence, we do not have to re-evaluate all intensity functions. Moreover, for exponential trigger kernels, each time we re-evaluate an intensity function, we exploit the properties of the exponential function to do so in $O(1)$: let t_1 and t_2 be two consecutive events, then, we can compute each intensity $\lambda^*_{up}(t_2)$ as $(\lambda^*_{up}(t_1) - \mu_p)\exp(-\omega(t_2 - t_1)) + \mu_p$.

The complete simulation algorithm is summarized in Algorithm 1, which needs $O(nd|\mathcal{V}|)$ operations to draw n samples, where d is the maximum number of neighbors per node. Note that, since the model parameters $(a_{lp})_{l,p\in\mathcal{P}}$ and $(b_{lp})_{l,p\in\mathcal{P}}$ may take negative values, it may happen that $\lambda^*_{up}(t) < 0$ for some time t. In those cases, as proposed by Ogata [42], we set $\lambda^*_{up}(t)$ to zero.

B. Properties of the simulated product adoption

By modeling recency and social influence, our model is able to generate *realistic* collections of product use events, which obey several temporal patterns observed in real social activity data [16], [17]. Here, we pay attention to two of these temporal patterns, which alternative models based on Poisson processes or Weibull renewal processes fail to capture:

- **Bursts.** Social events are characterized by bursts of rapidly occurring events separated by long periods of inactivity [16]. In our Twitter dataset (Refer to Section VI), we also find such bursts of events, for example, Figure 1(a) shows the times (in black) in which a user used url shortening services (*e.g.*, 'bit.ly'). Remarkably, our model (in red) is able to mimic such bursts of events while alternative models, such as Poisson processes (in blue), fail to do so. Here, we trained the parameters of both models using the user's real events (Refer to Section IV). Our model is able to generate bursts due to considering mutual excitation and inhibition between events in the intensities, given by Eq. 15.
- **Product Switches.** A person who uses a product may decide to switch to a competing product due to multiple factors, including social influence [17]. In our Twitter dataset (Refer to Section VI), we often find users that decide to switch products, for example, Figure 1(b) shows the times (in the center) in which a user switch from one retweet convention, 'RT' (in red), to another, 'via' (in blue). Our model (in the top) is able to generate such a product switch while a Poisson process (in the bottom) does not succeed. Again, we trained the parameters of both models using the user's real events (Refer to Section IV). Our model is able to generate product switches because it models competition by considering inhibition between events in the intensities, given by Eq. 15.

We remark that here our goal here is to show that our model can generate product use events obeying several temporal patterns observed in real social activity . However, since product uses are realizations of a (stochastic) continuous-time temporal point process, one cannot expect to precisely match the timing of each individual event.

Algorithm 1 Efficient Model Simulation

```
Initialization: $n^{up} = 0$ for $u = 1, \dots, |\mathcal{V}|, p = 1, \dots, |\mathcal{P}|$
1:  $I^* \leftarrow I^{|\mathcal{V}|\times|\mathcal{P}|}(t_0) \leftarrow \sum_u^{|\mathcal{V}|}\sum_p^{|\mathcal{P}|} \lambda_p^u(t_0)$
Generate first event:
2:  Generate $q \sim \mathcal{U}_{[0,1]}$ and $s \leftarrow t_0 - \frac{1}{I^*}\ln(q)$
3:  if $s > T$, then go to last step.
4:  else Attribution Test:
        i) Sample $d \sim \mathcal{U}_{[0,1]}$
        ii) Choose $u$ and $p$ such that $\frac{I^{up-1}(t_0)}{I^*} < d \leq \frac{I^{up}(t_0)}{I^*}$
        iii) Set $t_1 \leftarrow t_1^{up} \leftarrow s$, $i \leftarrow 1$ and $n^{up} \leftarrow 1$
General subroutine:
5:  while $s < T$ do
6:      $I^* \leftarrow I^{|\mathcal{V}|\times|\mathcal{P}|}(t_i) + \sum_{pl}^{|\mathcal{P}|} a_{pl}^u + \sum_{u'\in\mathcal{N}(u)} b_{pl}^{u'}$
7:      Generate $q \sim \mathcal{U}_{[0,1]}$
8:      Update $s \leftarrow t_i - \frac{1}{I^*}\ln(q)$
9:      if $s > T$, then go to last step
10:     else Attribution-Rejection Test:
            i) Sample $d \sim \mathcal{U}_{[0,1]}$
            ii) if $d \leq \frac{I^{|\mathcal{V}|\times|\mathcal{P}|}(s)}{I^*}$, then
                - Choose $u$ and $p$ such that
                      $\frac{I^{up-1}(t_0)}{I^*} < d \leq \frac{I^{up}(t_0)}{I^*}$
                - Set $t_{i+1} \leftarrow t_{n^{up}+1}^{up} \leftarrow s$, $i \leftarrow i+1$
                  and $n^{up} \leftarrow n^{up} + 1$
            iii) else
                - Update $I^* \leftarrow I^{|\mathcal{V}|\times|\mathcal{P}|}(s)$ and go to step 8.
    end while
11: Output: Retrieve the simulated process
    $(\{t_i^{up}\})_{u=1,\dots,|\mathcal{V}|,p=1,\dots,|\mathcal{P}|}$ on $[t_0, T]$
```

IV. Model Parameter Estimation

In this section, we show how to efficiently learn the parameters of our model using a set of historical product use events. Given a collection of product use events $\mathcal{H}(T) = \{(u_i, p_i, t_i)\}$ recorded during a time period $[0, T)$ in a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, our goal is to find the optimal parameters $\boldsymbol{\mu}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ by solving a regularized maximum likelihood estimation (MLE) problem. To this end, we first compute the log-likelihood of the recorded events, using Eq. 10, as follows:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}) = \sum_{e_i \in \mathcal{H}(T)} \log \lambda_{u_i p_i}^*(t_i) - \sum_{u \in \mathcal{G}} \sum_{p \in \mathcal{P}} \int_0^T \lambda_{up}^*(\tau)\, d\tau. \tag{16}$$

where we can further express the integral terms as:

$$\int_0^T \lambda_{up}^*(\tau)\, d\tau = T\mu_p + \sum_{l=1}^{|\mathcal{P}|} a_{lp} \sum_{e_i \in \mathcal{H}_{ul}(t)} G_\omega(T - t_i) + \sum_{v \in \mathcal{N}(u)} \sum_{l=1}^{|\mathcal{P}|} b_{lp} \sum_{e_i \in \mathcal{H}_{vl}(t)} G_\omega(T - t_i), \tag{17}$$

where $G_\omega(t) = \int_0^t g_\omega(t')dt'$.

Then, we can formulate the regularized MLE problem as:

$$\begin{aligned} \underset{\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}}{\text{minimize}} \quad & -\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}) + \beta\, \mathcal{R}(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}), \\ \text{subject to} \quad & \mu_p \geq 0, \end{aligned} \tag{18}$$

where the first term is the negative log-likelihood of the events and the second term is the regularization term, being β the parameter that controls the trade-off between these two terms. As long as we choose a convex regularizer, it is easy to show that this regularized MLE problem is jointly convex in $\boldsymbol{\mu}$, $\mathbf{A}$ and $\mathbf{B}$ by using linearity, composition rules for convexity, and the concavity property of the logarithm. Hence, the global optimum can be found by many well-known algorithms [13], [14]. In practice, we solved Eq. 18 with CVX, a software package for specifying and solving convex programs [44], and used a quadratic regularizer, $\mathcal{R}(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}) = \|\boldsymbol{\mu}\|_2^2 + \|\boldsymbol{A}\|_2^2 + \|\boldsymbol{B}\|_2^2$, which we found to work well in practice.

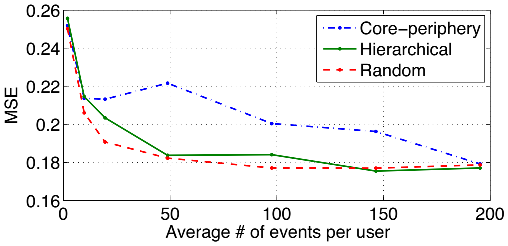


Fig. 2. Performance of our model estimation method in three Kronecker networks. We set the number of nodes in the three networks to 512 users and the number of edges to 2,040 (Core-Periphery), 4,608 (Hierarchical) and 7,669 (Random).

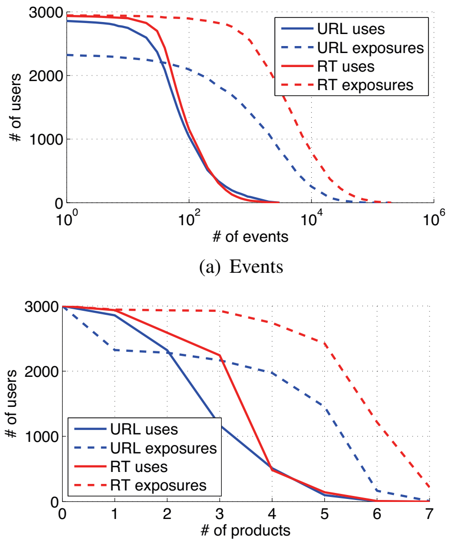
For simplicity, we have described the model using the same model parameters, $\boldsymbol{\mu}$, $\boldsymbol{A}$ and $\boldsymbol{B}$, for all users. However, in our experiments, we use different model parameters, $\boldsymbol{\mu}^u$, $\boldsymbol{A}^u$ and $\boldsymbol{B}^u$ and, by doing so, we need to fit $2|\mathcal{V}||\mathcal{P}|^2 + |\mathcal{V}||\mathcal{P}|$ parameters. Fortunately, we can decompose the model estimation procedure into $|\mathcal{V}| \times |\mathcal{P}|$ independent maximum likelihood estimation problems, one per user u and product p, on $2|\mathcal{P}|+1$ variables, and solve each subproblem in parallel, obtaining local solutions that are globally optimal. Additionally, under exponential kernels, we can precompute all sums of triggering kernels and integrals of triggering kernels for each subproblem in linear time, *i.e.*, $O(|\mathcal{H}_u(T)| + |\cup_{v \in \mathcal{N}(u)} \mathcal{H}_v(T)|)$, by exploiting the properties of the exponential function, similarly as described in Section III. As a result, our model estimation procedure scales to networks on the order of hundreds of thousands of nodes.

V. Experiments on synthetic data

In this section, we validate our model using synthetic networks and product use event data. In particular, we will demonstrate that our model estimation method can accurately recover the true model parameters from historical event data using synthetic data.

A. Experimental setup

We generate three types of synthetic networks using a well-known model of directed social networks, the Kronecker graph model [45]: (i) *core-periphery* networks [46] (parameter matrix: $[0.9, 0.5; 0.5, 0.3]$), (ii) *hierarchical* networks [47] ($[0.9, 0.1; 0.1, 0.9]$), and (iii) *random* networks ($[0.5, 0.5; 0.5, 0.5]$). Then, for each network, we assume there are two competing products, and set the model parameters for each product and node in the networks as follows. First, we draw $\{a_{pp}^u\}$ and $\{b_{pp}^u\}$ from $U(0, 1)$ and $\{a_{lp}^u\}_{l \neq p}$ and $\{b_{lp}^u\}_{l \neq p}$ from $U(-1, 1)$. Then, we allow for a small set of users to have a baseline parameter greater than zero, and



(a) Events

(b) Products

Fig. 3. Product use event statistics. Panel (a) shows the number of users that generated (solid lines) or got exposed to (dashed lines) at least x use events during the considered 4-month period. Panel (b) shows the number of users that used (solid lines) or got exposed to (dashed lines) at least x different products during the considered 4-month period.

draw their baseline parameters from the uniform distribution $U(0, 1)$. Finally, for each network, we generate and record a set of 100,000 events using our efficient sampling method, summarized in Algorithm 1. Now, given the times when each user used any of the two products, our goal is to find the true model parameters $\boldsymbol{\mu}^u$, $\mathbf{A}^u$, and $\mathbf{B}^u$ for each user u, by solving the regularized maximum likelihood estimation (MLE) problem defined in Eq. 18.

B. Model Estimation

We evaluate our model estimation method by comparing the inferred and true parameters in terms of the mean squared error (MSE), $E\left[(x - \hat{x})^2\right]$, where x is the true parameter and $\hat{x}$ is the estimated parameter. Figure 2 shows the MSE against average number of events per user for three Kronecker networks, where we set the regularization parameter $\beta = 10$ and the parameter of the triggering kernels $\omega = 1$. We observe that the MSE decreases as the average number of events per user increases, reaching values below 0.18 for the three networks once we observe ∼200 product use events per user in average. It is important to note that even though the networks have very different global network structures, the performance of our estimation method is remarkably stable and does not seem to depend on the structure of the network.

VI. Experiments on Real Data

In this section, we validate our model using real-world networks and product use event data. We first show that our model can accurately predict product use events in real-world data gathered from Twitter [18], significantly outperforming a state of the art method [7] and two baselines. We then use our model to derive insights into some of the factors driving product adoption and frequency of use, *i.e.*, social influence, recency and product competition.

(a) Average prediction probability across all users.

Data	Our model	Poisson	Weibull	Recency [7]
URL	**0.83**	0.64	0.45	0.78
RT	**0.82**	0.65	0.78	0.81

(b) Percentage of users for which each model achieves the maximum prediction probability.

Data	Our model	Poisson	Weibull	Recency [7]
URL	**81.6%**	32.8%	58.2%	41.3%
RT	**89.1%**	45.2%	76.3%	57.3%

TABLE I. Prediction Probability.

A. Experimental Setup

We use data gathered from Twitter as reported in previous work [18], which comprises of 1.7 billion public tweets posted by 52 million users during a three year period, from March 2006 to September 2009. Importantly, this dataset is complete, meaning that it contains all public tweets posted before the end of September 2009. Based on this raw data, we build two dataset of product use events. In the first dataset, we track every use event of the seven most popular (used) url shorteners [5]: Bitly, TinyURL, Isgd, TwURL, SnURL, Doiop and Eweri. Url shorteners existed before Twitter, however, by constraining the number of characters per message, Twitter increased the proliferation of new url shorteners as well as their usage. In the second dataset, we track every use event of the seven most popular retweet conventions[1] [19]: RT, via, HT, retweet, retweeting, the symbol ♻ and R/T. Retweet conventions emerged organically during the first few years of Twitter and until November 2009, when Twitter rolled out an official built-in retweet button. In both cases, the functionality (or quality) of each competing product is similar, therefore, one may expect social influence, recency and competition factors to play a more significant role than product quality [7]. This allows us to directly observe and quantify the impact of each of these factors in the behavior pattern of each user in the network. It would be also interesting to consider competing products with clearly different quality and augment our model to incorporate product quality as a covariate in the product intensity defined by Eq. 15.

For each of the two datasets, we experiment with four consecutive (arbitrary) months of product event data[2], where we filter out users that join Twitter less than one month before the start of the period, do not use any product after the period, or generate less than 100 product use events during the whole three years of data, and focus on the 3,000 most active users (within the period) and their neighbors. We build each user's neighborhood using the interactions via @-messages; we create a directed edge (i, j) as soon as user j mentions user i in a tweet, since this provides evidence that user j is paying attention to user i [48]. Then, we assume node i got exposed only to product use events from node j that occurred later than this first mention, as argued in previous work [49]. Here, one could think of building each user's neighborhood

[1] The way Twitter users indicated back in 2009 that a tweet was being retweeted (or forwarded). Here, we think of different retweet conventions as competing products.

[2] January 15 to May 15 for url shorteners; February 1 to May 31 for retweet conventions.

(a) Average test log-likelihood per event across all users.

Data	Our model	Poisson	Weibull
URL	**-1.09**	-3.28	-1.18
RT	**2.11**	-3.96	-1.76

(b) Percentage of users for which each model achieves the maximum average test log-likelihood per event.

Data	Our model	Poisson	Weibull
URL	**80.2%**	4.1%	15.7%
RT	**94%**	0%	6%

TABLE II. AVERAGE TEST LOG-LIKELIHOOD.

(a) Average AIC across all users. Lower is better.

Data	Our model	Poisson	Weibull
URL	**-430.4**	-6.4	-66.6
RT	**-44.6**	25.5	24.5

(b) Percentage of users for which each model achieves the maximum AIC.

Data	Our model	Poisson	Weibull
URL	**65%**	2.5%	32.5%
RT	**66.8%**	6.1%	27.1%

TABLE III. AKAIKE INFORMATION CRITERION (AIC).

by simply considering all the users she follows, however, this would require obtaining the times in which each user starts following others, and these times are currently not provided by the Twitter public API [50]. In the remainder of the section, for each dataset, we employ the first three months of data as training set, and the last month as test set.

Figure 3 provides general statistics on the number of events users generate and are exposed to through her neighbors in both datasets. We find that $\sim 35\%$ of the url shortener users and $\sim 75\%$ of the retweet convention users use three or more products. Moreover, retweet conventions users are exposed to a larger number of product use events and products than url shorteners users.

B. Implementation and scalability

We developed an efficient distributed implementation of our model estimation method, described in Section IV, and deployed it in a cluster with 1,000 CPU cores. This allowed us to efficiently model thousands of users. Although we considered five url shorteners and retweet conventions, in practice, due to the cross validation step of the regularization parameter β and the triggering kernel parameter ω, we experimented with 38 different configurations per product and user. Therefore, since we modeled 3,000 users for both url shortener and retweet conventions, the analysis involved estimating over 50 million model parameters by solving 1,140,000 ($3{,}000 \times 5 \times 38 \times 2$) convex optimization problems, in which the number of events in the objective function spans from 0 to hundreds of events. Remarkably, for each optimization problem, we can precompute the sums of triggering kernels and integrals of triggering kernels in a time linear in the number of events a user is exposed to, as discussed in Section IV. As a consequence, the overall fitting procedure for a very active user never took more than a few seconds and, for a less active user, it often took only a few milliseconds.

C. Product use event prediction

We evaluate the performance of our model in comparison with one state of the art method [7] ('Recency'), which is the only approach in the literature that deal with recurrent usages of products and not only for single adoptions, and therefore allows for fair comparison with ours. We also compare our model to two baselines, a memoryless Poisson process ('Poisson') [51] and a Weibull renewal process ('Weibull') [16]. The Recency model is a discrete time model that considers a product from t events ago to be reused with a probability proportional to a function of t. Poisson and Weibull are continuous time models that consider, respectively, a constant (memory-less) intensity, as defined in Eq. 5, and an intensity that depends on the last event, as defined in Eq. 6. For each user, we fit all models using maximum likelihood estimation. In our model, we set the regularization parameter β and the decay function parameter ω via cross-validation. In the Recency model, we consider a memory equal to five events.

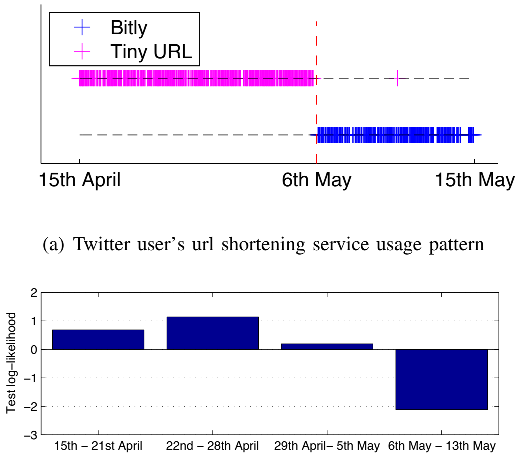
Then, we use three different evaluation measures to compare our model to the other three models:

I. **Prediction Probability:** For each user and (trained) model, we predict the identity of the product associated to each true event in her test set by evaluating and ranking the likelihoods (probability for the Recency model) associated to each product at the test event and taking the top-1 product. The prediction probability is then defined as the probability that the true and the predicted identity for each product use test event coincide. This evaluation measure has been used previously by the authors of the Recency method [7].

II. **Average Test Log-likelihood:** For each user and continuous-time (trained) model, we compute the average log-likelihood per event in the test set. This allows us to assess how well each continuous-time model fits the true times of each test event – in other words, its goodness of fit. Note that we compute the log-likelihood over the test data, not the training data, to measure how *well* each model generalizes to the data. Here, we cannot compute the average test log-likelihood for the discrete-time Recency model, since it does not model the actual time of each product use event.

III. **AIC:** For each user and continuous-time (trained) model, we compute the AIC [52] (Akaike information criterion). The AIC is given by $AIC = 2N_p - 2\mathcal{L}$, where N_p is the number of model parameters and $\mathcal{L}$ is the average log-likelihood evaluated in the training set. AIC quantifies the trade-off between the goodness of fit for the true times of each training event and the complexity of the model and thus has been typically used for model selection. The preferred model is the one with the minimum AIC value. As with the average test log-likelihood, we cannot compute the AIC for the discrete-time Recency model, since it does not model the actual time of each product use event.

Table I(a) shows the average product prediction probability across all users. Our model outperforms the Recency model



(a) Twitter user's url shortening service usage pattern

(b) Average test log-likelihood per event and user

Fig. 4. An example of external intervention. On May 6th, 2009, Twitter changed the default url shortener from 'tinyurl.com' to 'bit.ly'.

(a) URL shortening services

Product	$\mu_p^u > 0$	$a_{pp}^u > 0$	$b_{pp}^u > 0$
Bitly	100%	39.2%	39.4%
Tiny URL	100%	89.1%	82.3%
Isgd	0%	16.3%	23.6%
TwURL	0%	11.2%	19.9%
SnURL	0%	2.9%	3.6%

(b) Retweet conventions

Convention	$\mu_p^u > 0$	$a_{pp}^u > 0$	$b_{pp}^u > 0$
RT	20.5%	94.9%	89.2%
via	15%	59.6%	66.5%
HT	25.4%	60.2%	67.2%
retweet	11.1%	4%	7.5%
retweeting	3.9%	3.2%	4.8%

TABLE IV. PERCENTAGE OF USERS WITH PARAMETERS $\mu_p^u > 0$, $a_{pp}^u > 0$ AND $a_{pp}^u > 0$.

and both baselines for both url shorteners and retweet conventions. At first, one may think that the improvement with respect to the Recency model is small[3], however, Table I(b) demonstrates that our model consistently achieves the highest prediction probability for each individual user by showing the percentage of users for which each model provides the maximum product prediction probability across all models. In terms of individual users, our model beats the second best, the Weibull model, by more than 20% for url shorteners and more than 10% for retweet conventions while the Recency and the Poisson models achieve an underwhelming performance. Here, note that the percentages sum up more than 100% because whenever we encounter a user that generate a few use events during the test period, several models often provide the same maximum prediction probability.

Table II(a) shows the average test log-likelihood per event across users. Our model outperforms both continuous-time baseline models for url shorteners and retweet conventions. Moreover, Table II(b) demonstrate that our model consistently achieves the highest average log-likelihood across the continuous-time models for each individual user by showing the percentage of users for which each model provides the maximum test log-likelihood. In particular, our model is the best fit for as much as 80.2% and 92% of url shortening services and the retweet conventions users, respectively. Note that probability densities of continuous random variables can be larger than 1, and thus test log-likelihoods be larger than 0 (*e.g.*, our model achieves an average test log-likelihood value of 2.11 for retweet conventions).

Table III(a) shows the average AIC across users and Table III(b) the percentage of users for which each model provides the minimum AIC. Our model beats both continuous-time baseline models in more than 65% of the users. Hence, we can conclude that our model is not only the model with the best performance in terms of product prediction probability and test log-likelihood, but it also provides the best trade-off between goodness of fit and model complexity.

D. External interventions

We have shown the effectiveness of our model at predicting the adoption and frequency of use of competing products. However, what if there is a *sudden* exogenous change, which cannot be explained through social influence, recency and product competition? For example, on May 6th, 2009, Twitter changed the default url shortener from 'tinyurl.com' to 'bit.ly', triggering a significant increase (decrease) on the amount of new adopters of 'bit.ly' ('tinyurl.com') shortly afterwards, as illustrated by Fig. 4(a). Can we at least detect such a change once it happened? As Figure 4(b) demonstrates for the above mention example, such change is easily identifiable by means of the average test log-likelihood per event across users, which drops dramatically. Therefore, once a change is detected, by means of the average log-likelihood per event, one may decide to re-train the model.

E. Social influence, recency and competition

We conclude our experiments by exploring the properties of the inferred users' parameters on real data both qualitatively and quantitatively. By doing so, we aim to gain insights into some of the factors driving adoption and frequency of use of competing products. Here, we focus on the five most popular url shorteners and retweet conventions, since the parameters of the two less popular shorteners and retweet conventions are hardly ever different than zero.

We start by comparing spontaneous adoption (by means of μ_p^u), recency (a_{pp}^u) and (positive) social influence (b_{pp}^u) on the adoption and frequency of use of a product. Table IV summarizes the results, by showing the percentage of users for which μ_p^u, a_{pp}^u and b_{pp}^u are greater than zero for url shorteners and retweet conventions. We find clear differences between url shorteners and retweet conventions, which we summarize next. On the one hand, Table IV(a) indicates that, for the 3,000 users under study, spontaneous adoption dominates over recency or (positive) social influence to explain the use of popular shorteners ('bitly' and 'tinyurl'). In contrast, less popular shorteners ('isgd', 'twurl' and 'snurl') are never used by spontaneous adoption but first used by (positive) social

[3] The difference is statistically significant (Wilcoxon rank sum test, $p = 0.034$).

(a) URL shortening services

p l	Bitly	Tiny URL	Isgd	TwURL	SnURL
Bitly		58.8%	9.7%	7.2%	1.3%
Tiny URL	2.64%		1.1%	0.5%	0.2%
Isgd	28.2%	80.8%		9.2%	1.4%
TwURL	33%	81.8%	21.6%		2.8%
SnURL	36.6%	87.9%	24.6%	13.8%	

(b) Retweet conventions

p l	RT	via	HT	Retweet	Retweeting
RT		1.6%	1%	0.1%	0.1%
via	25.8%		10.7%	3.3%	2.6%
HT	22.2%	12.9%		2.5%	2.3%
Retweet	52%	45.2%	40%		1.6%
Retweeting	54.3%	62%	62.5%	4.4%	

TABLE V. PERCENTAGE OF USERS WITH PARAMETERS $a^u_{lp} < 0$.

(a) URL shortening services

p l	Bitly	Tiny URL	Isgd	TwURL	SnURL
Bitly		30.6%	2.6%	4.6%	0.6%
Tiny URL	6.5%		1.8%	4%	0.3%
Isgd	30.1%	86.4%		7.2%	0.9%
TwURL	29.7%	82.6%	15.6%		1.6%
SnURL	31.5%	81.9%	17.9%	10.5%	

(b) Retweet convention

p l	RT	via	HT	Retweet	Retweeting
RT		2.1%	1%	0.2%	0.1%
via	27.6%		1.8%	0.6%	0.6%
HT	26.5%	1.3%		0.5%	0.6%
Retweet	56.6%	47.2%	19.6%		1.6%
Retweeting	64.3%	61.1%	55.5%	2.6%	

TABLE VI. PERCENTAGE OF USERS WITH PARAMETERS $b^u_{lp} < 0$.

influence and then re-used by recency. On the other hand, Table IV(b) indicates that recency and (positive) influence dominate over spontaneous adoption to explain the use of popular retweet conventions ('RT', 'via' and 'HT'). In contrast, spontaneous adoption, social influence and recency have a comparable strength on the adoption and use of less popular retweet conventions ('retweet' and 'retweeting').

Next, we investigate competition (or inhibition) between products. Tables V and VI summarize the results, by showing the percentage of users for which a^u_{lp} and b^u_{lp} (for $l \neq p$) are smaller than zero. Here, each value can be viewed as a measure of the degree of inhibition that an event of the product in the row entails on the usage of the products in each column. We find several interesting patterns, which we summarize next. First, if we pay attention to the last rows of each table, which correspond to less popular products and conventions, we conclude that using one of these products or conventions has a strong inhibitory effect on using more popular product or conventions. Second, we find that 'bit.ly' has a stronger inhibitory effect on 'tinyurl.com' but 'tinyurl.com' does not have almost any effect on 'bit.ly'. This may indicate that even before the change of default service from 'tinyurl.com' to 'bit.ly' on May 6th, 'bit.ly' was already a strong competitor to 'tinyurl.com'. Finally, if we compare the inhibitory effect of previous product events by a user herself (a^u_{lp}) and her neighbors (b^u_{lp}), we find qualitatively similar values.

VII. Conclusions

We have developed a probabilistic model, based on temporal point processes, for the adoption and frequency of use of competing products in online social networks. By modeling recency, social influence and product competition, our model is able to generate realistic collections of product use events, obeying several temporal patterns observed in real online social activity, as well as predict product use events more accurately than alternatives.

Our work also opens many interesting venues for future work. For example, we have modeled the decay of the influence of an event over time using exponential triggering kernels $g_\omega(\cdot)$ and considered linear intensities with constant baseline rates. However, this is implicitly considering one characteristic time scale, which may be unrealistic in some scenarios. A natural follow-up to potentially improve the accuracy of our model would be considering more complex triggering kernels [14], performing a non-parametric estimation of the kernels [43], assuming time-varying baselines rates [39] or extending the model to incorporate nonlinearities [53]. We have considered considered products and contagions with similar functionalities (or qualities) in order to study the impact of social influence, recency and competition play in the users' decisions. However, as future work, it would be also interesting to consider competing products with clearly different quality and augment our model to explicitly incorporate product quality. Moreover, we have built each user's neighborhood using the interactions via @-messages and considered each neighbor to have the same influence on a user. It would be interesting to build each user's neighborhood using the actual follower network and consider each neighbor to have different influence[4]. Although our experimental validation relies on data gathered exclusively from Twitter, it would be interesting to evaluate our model in other microblogging (Weibo, Tumblr) and social networking (Facebook, G+) sites. Finally, we have assumed that all product uses and users' ego-networks are observable. It would be very interesting to relax these assumptions and apply our model to a wide spectrum of products, from competing operative systems (*e.g.*, Windows, MacOS and Linux) to competing beers (*e.g.*, Corona, Heineken, etc.).

References

[1] V. Mahajan, E. Muller, and F. M. Bass, "New product diffusion models in marketing: A review and directions for research," *The Journal of Marketing*, 1990.

[2] F. M. Bass, "A new product growth for model consumer durables," *Management Science*, vol. 15, no. 1, p. 215, 1969.

[3] D. McFadden, "Econometric models for probabilistic choice among products," *Journal of Business*, vol. 53, no. 3, pp. S13–S29, 1980.

[4] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," *The Journal of Economic Perspectives*, vol. 12, no. 3, pp. 151–170, 1998.

[4] This would require obtaining the times in which each user starts following others and, unfortunately, these times are currently not provided by the Twitter public API [50].

[5] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis, "we. b: The web of short urls," in *20th International Conference on World Wide Web*, 2011.

[6] D. Kempe, J. M. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[7] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii, "The dynamics of repeat consumption," in *23rd International Conference on World Wide Web*, 2014.

[8] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," *Internet and Network Economics*, pp. 306–311, 2007.

[9] T. Liniger, "Multivariate hawkes processes," Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, 2009.

[10] C. Blundell, J. Beck, and K. A. Heller, "Modelling reciprocating relationships with hawkes processes," in *Neural Information Processing Systems*, 2012.

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

[12] T. Iwata, A. Shah, and Z. Ghahramani, "Discovering latent influence in online social activities via shared cascade poisson processes," in *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[13] K. Zhou, L. Song, and H. Zha, "Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes," in *16th International Conference on Artificial Intelligence and Statistics*, 2013.

[14] K. Zhou, H. Zha, and L. Song, "Learning triggering kernels for multi-dimensional hawkes processes," in *30th International Conference on Machine Learning*, 2013.

[15] Y. Ogata, "On lewis' simulation method for point processes," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 23–31, 1981.

[16] A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, p. 207, 2005.

[17] P. Shukla, "Effect of product usage, satisfaction and involvement on brand switching behaviour," *Asia Pacific Journal of Marketing and Logistics*, vol. 16, no. 4, pp. 82–104, 2004.

[18] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring User Influence in Twitter: The Million Follower Fallacy," in *4th AAAI Conference on Weblogs and Social Media*, 2010.

[19] F. Kooti, H. Yang, M. Cha, P. K. Gummadi, and W. A. Mason, "The emergence of conventions in online social networks." in *6th AAAI Conference on Weblogs and Social Media*, 2012.

[20] P. Dubey, B. De Meyer, and R. Garg, *Competing for customers in a social network*. Cowles Foundation Discussion Paper, 2006.

[21] S. Goyal and M. Kearns, "Competitive contagion in networks," in *44th ACM Symposium on Theory of Computing*, 2012.

[22] S. Myers and J. Leskovec, "Clash of the Contagions: Cooperation and Competition in Information Diffusion," in *12th IEEE International Conference on Data Mining*, 2012.

[23] J. Kostka, Y. A. Oswald, and R. Wattenhofer, "Word of mouth: Rumor dissemination in social networks," in *SIROCCO*, 2008.

[24] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the Spread of Misinformation in Social Networks," in *4th ACM International Conference on Web Search and Data Mining*, 2011.

[25] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[26] N. E. Friedkin and E. C. Johnsen, "Social influence and opinions," *Journal of Mathematical Sociology*, vol. 15, no. 3-4, pp. 193–206, 1990.

[27] A. Beutel, B. A. Prakash, R. Rosenfeld, and C. Faloutsos, "Interacting viruses in networks: can both survive?" in *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[28] B. A. Prakash, A. Beutel, R. Rosenfeld, and C. Faloutsos, "Winner takes all: competing viruses or ideas on fair-play networks," in *21st International Conference on World Wide Web*, 2012.

[29] N. Du, L. Song, A. Smola, and M. Yuan, "Learning networks of heterogeneous influence," in *Neural Information Processing Systems*, 2012.

[30] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the Temporal Dynamics of Diffusion Networks," in *28th International Conference on Machine Learning*, 2011.

[31] J. Teevan, E. Adar, R. Jones, and M. Potts, "History repeats itself: repeat queries in yahoo's logs," in *29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[32] J. Teevan, E. Adar, R. Jones, and M. A. Potts, "Information re-retrieval: repeat queries in yahoo's logs," in *30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[33] E. Adar, J. Teevan, and S. T. Dumais, "Large scale analysis of web revisitation patterns," in *SIGCHI conference on Human Factors in Computing Systems*, 2008.

[34] ——, "Resonance on the web: web dynamics and revisitation patterns," in *SIGCHI Conference on Human Factors in Computing Systems*, 2009.

[35] D. Marsan and O. Lengline, "Extending earthquakes' reach through cascading," *Science*, vol. 319, no. 5866, pp. 1076–1079, 2008.

[36] M. Egesdal, C. Fathauer, K. Louie, J. Neuman, G. Mohler, and E. Lewis, "Statistical modeling of gang violence in los angeles," *SIAM Undergraduate Research Online*, vol. 3, 2010.

[37] M. Krumin, I. Reutsky, and S. Shoham, "Correlation-based analysis and generation of multiple spike trains using hawkes models with an exogenous input," *Frontiers in computational neuroscience*, 2010.

[38] E. Bacry, S. Delattre, M. Hoffmann, and J.-F. Muzy, "Modelling microstructure noise with mutually exciting point processes," *Quantitative Finance*, vol. 13, no. 1, 2013.

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

[40] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Neural Information Processing Systems*, 2013.

[41] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling Information Propagation with Survival Theory," in *30th International Conference on Machine Learning*, 2013.

[42] Y. Ogata and H. Akaike, "On linear intensity models for mixed doubly stochastic poisson and self-exciting point processes," in *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 269–274.

[43] E. Bacry, K. Dayri, and J.-F. Muzy, "Non-parametric kernel estimation for symmetric hawkes processes. application to high frequency financial data," *The European Physical Journal B*, vol. 85, no. 5, pp. 1–12, 2012.

[44] M. Grant and S. Boyd, "CVX," http://cvxr.com/cvx, 2013.

[45] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, 2010.

[46] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *17th International Conference on World Wide Web*, 2008.

[47] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[48] B. A. Huberman, D. M. Romero, and F. Wu, "Social networks that matter: Twitter under the microscope," *First Monday*, vol. 14, no. 1, 2009.

[49] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter," in *20th International Conference on World Wide Web*, 2011.

[50] S. A. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *23rd International Conference on World Wide Web*, 2014, pp. 913–924.

[51] J. Kingman, *Poisson Processes*, ser. Oxford studies in probability. Clarendon Press, 1992.

[52] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.

[53] P. Brémaud and L. Massoulié, "Stability of nonlinear hawkes processes," *The Annals of Probability*, pp. 1563–1588, 1996.