

RedQueen: An Online Algorithm for Smart Broadcasting in Social Networks

Ali Zarezade*
Sharif Univ. of Tech.
zarezade@ce.sharif.edu

Utkarsh Upadhyay*
MPI-SWS
utkarshu@mpi-sws.org

Hamid R. Rabiee
Sharif Univ. of Tech.
rabiee@sharif.edu

Manuel Gomez-Rodriguez
MPI-SWS
manuelgr@mpi-sws.org

ABSTRACT

Users in social networks whose posts stay at the top of their followers' feeds the longest time are more likely to be *noticed*. Can we design an online algorithm to help them decide when to post to stay at the top? In this paper, we address this question as a novel optimal control problem for jump stochastic differential equations. For a wide variety of feed dynamics, we show that the optimal broadcasting intensity for any user is surprisingly simple – it is given by the position of her most recent post on each of her follower's feeds. As a consequence, we are able to develop a simple and highly efficient online algorithm, REDQUEEN, to sample the optimal times for the user to post. Experiments on both synthetic and real data gathered from Twitter show that our algorithm is able to consistently make a user's posts more visible over time, is robust to volume changes on her followers' feeds, and significantly outperforms the state of the art.

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

General Terms: Algorithms; Experimentation.

Keywords: Smart broadcasting; Stochastic Optimal Control; Temporal Point Processes; Social Networks; Twitter.

1. INTRODUCTION

Whenever a user in an online social network decides to share a new story with her followers, she is often competing for *attention* with dozens, if not hundreds, of stories *simultaneously* shared by other users that their followers follow [2, 12]. In this context, recent empirical studies have shown that stories at the top of their followers' feed are more likely to be *noticed* and consequently liked or shared [15, 16, 21]. Can

*Authors contributed equally. This work was done during Ali Zarezade's internship at MPI-SWS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

© 2017 ACM. ISBN 978-1-4503-4675-7/17/02...\$15.00

DOI: <http://dx.doi.org/10.1145/3018661.3018684>

we find an algorithm that helps a user decide when to post to increase her chances to stay at the top?

The “when-to-post” problem was first studied by Spasojevic et al. [25], who performed a large empirical study on the best times to post in Twitter and Facebook, measuring attention a user elicits by means of the number of responses to her posts. Moreover, they designed several heuristics to pinpoint at the times that elicit the greatest attention in a training set and showed that these times also lead to more responses in a held-out set. Since then, algorithmic approaches to the “when-to-post” problem with provable guarantees have been largely lacking. Only very recently, Karimi et al. [17] introduced a convex optimization framework to find optimal broadcasting strategies, measuring attention a user elicits as the time that at least one of her posts is among the k most recent stories received in her followers' feed. However, their algorithm requires expensive data pre-processing, it does not adapt to changes in the users' feeds dynamics, and in practice, it is less effective than our proposed algorithm, as shown in Section 5.

In this paper, we design a novel online algorithm for the when-to-post problem, where we measure *visibility* of a broadcaster as the position of her most recent post on her followers' feeds over time. A desirable property of this visibility measure is that it can be easily extracted from real data without actual interventions — given any particular broadcasting strategy for a user, one can always measure its visibility using a separate held-out set of the user's followers' feeds [17]. In contrast, measures based on users' reactions (*e.g.*, number of likes, shares and replies) are difficult to estimate from real data, due to the presence of other confounding factors such as users' influence, content and wording [7, 8, 20].

More precisely, we represent users' posts and feeds using the framework of temporal point processes, which characterizes the continuous time interval between posts using conditional intensity functions [1]. Under this representation, finding the optimal broadcasting (or posting) strategy for a user reduces to finding its associated conditional broadcasting intensity [17]. Then, for a large family of intensity functions, which includes Hawkes [14] and Poisson [19] as particular instances, we find “when-to-post” by solving a novel optimal control problem for a system of jump stochastic differential equations (SDEs) [13]. Our problem formulation differs from previous literature in two key technical aspects, which are of independent interest:

- I. The control signal is a conditional (broadcasting) intensity, which is used to sample stochastic events (*i.e.*, stories to post). As a consequence, the problem formulation requires another layer of stochasticity. In previous work, the control signal is a time-varying real vector.
- II. The (broadcasting) intensities are stochastic Markov processes and thus the dynamics are doubly stochastic. This requires us to redefine the cost-to-go to incorporate the instantaneous value of these intensities as additional arguments. Previous work has typically considered constant intensities and only very recently time-varying deterministic intensities [27].

These technical aspects have implications beyond the smart broadcasting problem since they enable us to establish a previously unexplored connection between optimal control of jump SDEs and doubly stochastic temporal point processes (*e.g.*, Hawkes processes), which have been increasingly used to model social activity [10, 11, 28, 29].

Moreover, we find that the solution to the above optimal control problem is surprisingly simple: the optimal broadcasting intensity for a user is given by the position of her most recent post on each of her follower’s feeds. This solution allows for a simple and highly efficient online procedure to sample the optimal times for a user to broadcast, which can be implemented in a few lines of code and does not require fitting a model for the feeds’ intensities. Finally, we performed experiments on both synthetic and real data gathered from Twitter and show that our algorithm is able to consistently make a user’s posts stay at the top of her followers’ feeds, is robust to changes on the dynamics (or volume) of her followers’ feeds, and significantly outperforms the state of the art [17].

Further related work. In addition to the paucity of work on the when-to-post problem [17, 25], discussed previously, our work also relates to: (i) empirical studies on attention and information overload in social and information networks [2, 12, 15, 22], which investigate whether there is a limit on the amount of ties (*e.g.*, friends, followees or phone contacts) people can maintain, how people distribute attention across them, and how attention influences the propagation of information; and (ii) the influence maximization problem [6, 9, 10, 18, 24], which aims to find a set of nodes in a social network whose initial adoption of a certain idea or product can trigger the largest expected number of follow-ups. In contrast, we focus on optimizing a social media user’s broadcasting strategy to capture the greatest attention from the followers.

2. PRELIMINARIES

We first revisit the framework of temporal point processes [1] and then use it to represent broadcasters and feeds in social and information networks.

Temporal point processes. A temporal point process is a stochastic process whose realization consists of a sequence of discrete events localized in time, $\mathcal{H} = \{t_i \in \mathbb{R}^+ | i \in \mathbb{N}^+, t_i < t_{i+1}\}$. In recent years, they have been used to represent many different types of event data produced in online social networks and the web, such as the times of tweets [10], retweets [29], or links [11].

A temporal point process can also be represented as a counting process $N(t)$, which is the number of events up

to time t . Moreover, given $\mathcal{H}(t) = \{t_i \in \mathcal{H} | t_i < t\}$, the history of event times up to but not including time t , we can characterize the counting process using the conditional intensity function $\lambda^*(t)$, which is the conditional probability of observing an event in an infinitesimal window $[t, t + dt)$ given the history $\mathcal{H}(t)$, *i.e.*,

$$\lambda^*(t)dt = \mathbb{P}\{\text{event in } [t, t + dt) | \mathcal{H}(t)\} = \mathbb{E}[dN(t) | \mathcal{H}(t)],$$

where $dN(t) := N(t + dt) - N(t) \in \{0, 1\}$ and the superscript * means that the intensity may depend on the history $\mathcal{H}(t)$.

The functional form for the intensity is often chosen to capture the phenomena of interest. For example, in the context of modeling social activity, retweets have been modeled using Hawkes processes [10, 29] and daily and weekly variations on the volume of posted tweets have been captured using Poisson processes [23, 17]. In this work, we consider the following general functional form, which includes Hawkes and Poisson as particular instances:

$$\lambda^*(t) = \lambda_0(t) + \alpha \int_0^t g(t-s)dN(s), \quad (1)$$

where $\lambda_0(t) \geq 0$ is a time-varying function, which models the publication of messages by users on their own initiative, the second term, with $\alpha \geq 0$, models the publication of additional messages (*e.g.*, replies, shares) by the users due to the influence that previous messages (their own as well as the ones posted by others) have on their intensity, and $g(t)$ denotes an exponential triggering kernel $e^{-wt}\mathbb{1}(t \geq 0)$. The second term makes the intensity dependent on history and a stochastic process by itself. Finally, the following alternative representation will be useful to design our stochastic optimal control algorithm for smart broadcasting (proven in the Appendix):

PROPOSITION 1. *Let $N(t)$ be a counting process with an associated intensity $\lambda^*(t)$ given by Eq. 1. Then the tuple $(N(t), \lambda^*(t))$ is a doubly stochastic Markov process, whose dynamics can be defined by the following jump SDE:*

$$d\lambda^*(t) = [\lambda_0'(t) + w\lambda_0(t) - w\lambda^*(t)]dt + \alpha dN(t), \quad (2)$$

with initial condition $\lambda^*(0) = \lambda_0(0)$.

In the remainder of the paper, to simplify the notation, we drop the superscript * from the intensities.

Representation of broadcasters and feeds. Given a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ users, we assume any user can be a broadcaster, a follower or both, each broadcaster can be followed by multiple followers, and each follower can follow multiple broadcasters. Then, we represent the broadcasting times of the users as a collection of counting processes denoted by a vector $\mathbf{N}(t)$, in which the i -th dimension, $N_i(t)$, is the number of messages or *stories* broadcasted by user i up to time t . Here, we denote the history of times of the stories broadcasted by user i by time t as $\mathcal{H}_i(t)$, the entire history of times as $\mathcal{H}(t) = \cup_{i \in \mathcal{V}} \mathcal{H}_i(t)$, and characterize these counting processes using their corresponding intensities, *i.e.*, $\mathbb{E}[d\mathbf{N}(t) | \mathcal{H}(t)] = \boldsymbol{\mu}(t) dt$.

Given the adjacency matrix $A \in \{0, 1\}^{n \times n}$, where $A_{ij} = 1$ indicates that user j follows user i , we can represent the times of the stories users receive in their feeds from the broadcasters they follow as a sum of counting processes, $A^T \mathbf{N}(t)$, and calculate the corresponding conditional intensities as $\boldsymbol{\gamma}(t) =$

$A^T \boldsymbol{\mu}(t)$. Here, we denote the history of times of the stories received by user j by time t as $\mathcal{F}_j(t) := \cup_{i \in \mathcal{N}(j)} \mathcal{H}_i(t)$, where $\mathcal{N}(j)$ is the set of users that j follows.

Finally, from the perspective of broadcaster i , it is useful to define the counting processes $\mathbf{M}_{\setminus i}(t) = A^T \mathbf{N}(t) - A_i N_i(t)$, in which the j -th dimension, $M_{j \setminus i}(t)$, represents the times of the stories user j receives due to other broadcasters she follows, and A_i is the i -th row of the adjacency matrix A . Moreover, for each of these counting processes, the conditional intensity is given by $\gamma_{j \setminus i}(t) = \gamma_j(t) - \mu_i(t)$ and the history is given by $\mathcal{F}_{j \setminus i}(t) := \mathcal{F}_j(t) \setminus \mathcal{H}_i(t)$.

3. PROBLEM FORMULATION

In this section, we first define our visibility measure, $r(t)$, then derive a jump stochastic differential equation that links our measure to the counting processes associated to a broadcaster and her followers, and conclude with a statement of the when-to-post problem for our visibility measure.

Definition of visibility. Given a broadcaster i and one of her followers j , we define the visibility function $r_{ij}(t)$ as the position or *rank* of the most recent story posted by i in j 's feed by time t , which clearly depends on the feed ranking mechanism in the corresponding social network. Here, for simplicity, we assume each user's feed ranks stories in inverse chronological order¹. However, our framework can be easily extended to any feed ranking mechanisms, as long as its rank dynamics can be expressed as a jump SDE².

Under the inverse chronological ordering assumption, position is simply the number of stories that other broadcasters posted in j 's feed from the time of the most recent story posted by i until t . Then, when a new story arrives to a user's feed, it appears at the top of the feed and the other stories are shifted down by one. If we identify the time of the most recent message posted by i by time t as $\tau_i(t) = \max\{t_k \in \mathcal{H}_i(t)\}$, then the visibility is formally defined as:

$$r_{ij}(t) = M_{j \setminus i}(t) - M_{j \setminus i}(\tau_i(t)), \quad (3)$$

Note that, if the last story posted by i is at the top of j 's feed at time t , then $r_{ij}(t) = 0$.

Dynamics of visibility. Given a broadcaster i with broadcasting counting process $N_i(t)$ and one of her followers j with feed counting process due to other broadcasters $M_{j \setminus i}(t)$, the rank of i in j 's feed $r_{ij}(t)$ satisfies the following equation:

$$r_{ij}(t + dt) = \underbrace{(r_{ij}(t) + 1)dM_{j \setminus i}(t)(1 - dN_i(t))}_{1. \text{ Increases by one}} + \underbrace{0}_{2. \text{ Becomes zero}} + \underbrace{r_{ij}(t)(1 - dM_{j \setminus i}(t))(1 - dN_i(t))}_{3. \text{ Remains the same}},$$

where each term models one of the three possible situations:

1. The other broadcasters post a story in $(t, t + dt]$, $dM_{j \setminus i}(t) = 1$, and broadcaster i does not post, $dN_i(t) = 0$. The position of the last story posted by i in j 's feed steps down by one, *i.e.*, $r_{ij}(t + dt) = r_{ij}(t) + 1$.
2. Broadcaster i posts a story in $(t, t + dt]$, $dN_i(t) = 1$, and the other broadcasters do not, $dM_{j \setminus i}(t) = 0$. No

¹At the time of writing, Twitter and Weibo rank stories in inverse chronological order by default and Facebook allows choosing such an ordering.

²This would require either having access to the corresponding feed ranking mechanism or reverse engineering it, which is out of the scope of this work.

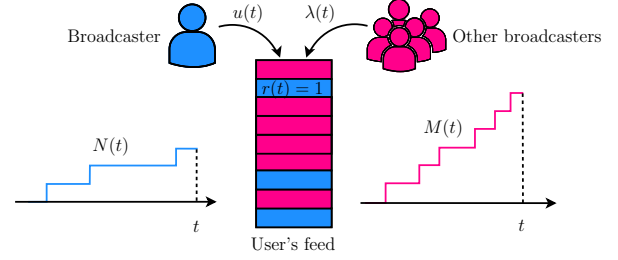


Figure 1: The dynamics of visibility. A broadcaster i posts $N_i(t) = N(t)$ messages with intensity $\mu_i(t) = u(t)$. Her messages accumulate in her follower j 's feed, competing for attention with $M_{j \setminus i}(t) = M(t)$ other messages, posted by other broadcasters j follows with intensity $\gamma_{j \setminus i}(t) = \lambda(t)$. The visibility function $r_{ij}(t) = r(t)$ is the position or rank of the most recent story posted by broadcaster i in the follower j 's feed by time t .

matter what the previous rank was, the new rank is $r_{ij}(t + dt) = 0$ since the newly posted story appears at the top of j 's feed.

3. No one posts any story in $(t, t + dt]$, $dN_i(t) = 0$ and $dM_{j \setminus i}(t) = 0$. The rank remains the same, *i.e.*, $r_{ij}(t + dt) = r_{ij}(t)$

We skip the case in which $M_{j \setminus i}(t) = 1$ and $dN_i(t) = 1$ in the same time interval $(t, t + dt]$ because, by the Blumenthal zero-one law [3], it has zero probability. Now, by rearranging terms and using $dN_i(t)dM_{j \setminus i}(t) = 0$, we uncover the following jump SDE for the visibility (or rank) dynamics:

$$dr_{ij}(t) = -r_{ij}(t)dN_i(t) + dM_{j \setminus i}(t) \quad (4)$$

where $dr_{ij}(t) = r_{ij}(t + dt) - r_{ij}(t)$. Figure 1 illustrates the concept of visibility for one broadcaster and one follower.

The when-to-post problem. Given a broadcaster i and her followers $\mathcal{N}(i)$, our goal is to find the optimal conditional intensity $\mu_i(t) = u(t)$ that minimizes the expected value of a particular nondecreasing convex loss function $\ell(\mathbf{r}(t), u(t))$ of the broadcaster's visibility on each of her follower's feed, $\mathbf{r}(t) = [r_{ij}(t)]_{j \in \mathcal{N}(i)}$, and the intensity itself, $u(t)$, over a time window $(t_0, t_f]$, *i.e.*,

$$\begin{aligned} & \underset{u(t_0, t_f]}{\text{minimize}} \quad \mathbb{E}_{(N_i, \mathbf{M}_{\setminus i})(t_0, t_f]} \left[\phi(\mathbf{r}(t_f)) + \int_{t_0}^{t_f} \ell(\mathbf{r}(\tau), u(\tau)) d\tau \right] \\ & \text{subject to} \quad u(t) \geq 0 \quad \forall t \in (t_0, t_f], \end{aligned} \quad (5)$$

where $u(t_0, t_f]$ denotes user i 's intensity from t_0 to t_f , the expectation is taken over all possible realizations of the counting processes associated to user i and all other broadcasters from t_0 to t_f , denoted as $(N_i, \mathbf{M}_{\setminus i})(t_0, t_f]$, and $\phi(\mathbf{r}(t_f))$ is an arbitrary penalty function³. Here, by considering a nondecreasing loss, we penalize times when the position of the most recent story on each of the follower's feeds is high (*i.e.*, the most recent story does not stay *at the top*) and we limit the number of stories the broadcaster can post. Finally, note that the optimal intensity $u(t)$ for broadcaster i at time t may depend on the visibility $\mathbf{r}(t)$ with respect to each of her

³The final penalty function $\phi(\mathbf{r}(t_f))$ is necessary to derive the optimal intensity $u^*(t)$ in Section 4. However, the actual optimal intensity $u^*(t)$ does not depend on the particular choice of terminal condition.

followers and thus the associated counting process $N_i(t)$ may be doubly stochastic.

4. STOCHASTIC OPTIMAL CONTROL ALGORITHM

In this section, we tackle the when-to-post problem defined by Eq. 5 from the perspective of stochastic optimal control of jump SDEs [13]. More specifically, we first derive a solution to the problem considering only one follower, provide an efficient practical implementation of the solution and then generalize it to the case of multiple followers. We conclude this section by deriving a solution to the problem given an (idealized) oracle that knows the times of all stories in the followers' feeds *a priori*, which we will use as baseline.

Optimizing for one follower. Given a broadcaster i with $N_i(t) = N(t)$ and $\mu_i(t) = u(t)$ and only one of her followers j with $M_{j \setminus i}(t) = M(t)$ and $\gamma_{j \setminus i}(t) = \lambda(t)$, we can rewrite the when-to-post problem defined by Eq. 5 as

$$\begin{aligned} & \underset{u(t_0, t_f]}{\text{minimize}} \quad \mathbb{E}_{(N, M)(t_0, t_f]} \left[\phi(r(t_f)) + \int_{t_0}^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \\ & \text{subject to} \quad u(t) \geq 0 \quad \forall t \in (t_0, t_f], \end{aligned} \quad (6)$$

where, using Eq. 2 and Eq. 4, the dynamics of $M(t)$ and $r(t)$ are given by the following two coupled jump SDEs:

$$\begin{aligned} dr(t) &= -r(t) dN(t) + dM(t) \\ d\lambda(t) &= [\lambda'_0(t) + w\lambda_0(t) - w\lambda(t)] dt + \alpha dM(t), \end{aligned}$$

with initial conditions $r(t_0) = r_0$ and $\lambda(t_0) = \lambda_0$, and the dynamics of $N(t)$ are given by the intensity $u(t)$ that we aim to optimize. The above stochastic optimal control problem differs from previous literature in two key technical aspects, which require careful reasoning:

- (i) The control signal $u(t)$ is a conditional intensity, which controls the dynamics of the counting process $N(t)$ (*i.e.*, number of stories broadcasted by user i by time t). As a consequence, the problem formulation needs to account for another layer of stochasticity. Previous work assumes the control signal to be a time-varying real vector.
- (ii) The dynamics of the counting process $M(t)$ (*i.e.*, number of stories broadcasted by other users that j follows by time t) are doubly stochastic Markov. Previous work has typically considered memoryless Poisson processes and, only very recently, inhomogeneous Poisson [27].

Next, we will define a novel optimal cost-to-go function that accounts for the above unique aspects of our problem, showing that the Bellman's principle of optimality still follows, and finally find the optimal solution using the corresponding Hamilton-Jacobi-Bellman (HJB) equation.

DEFINITION 2. *The optimal cost-to-go $J(r(t), \lambda(t), t)$ is defined as the minimum of the expected value of the cost of going from state $r(t)$ with intensity $\lambda(t)$ at time t to final state at time t_f , *i.e.*,*

$$\min_{u(t, t_f]} \mathbb{E}_{(N, M)(t, t_f]} \left[\phi(r(t_f)) + \int_t^{t_f} \ell(r(\tau), u(\tau)) d\tau \right], \quad (7)$$

where the expectation is taken over all trajectories of the control and noise jump processes, N and M , in the $(t, t_f]$ interval, given the initial values of $r(t)$, $\lambda(t)$ and $u(t)$.

Algorithm 1: REDQUEEN for fixed s , q and one follower.

Input: Parameters q and s
Output: Returns time for the next post
 $t \leftarrow \infty$; $\tau \leftarrow \text{othersNextPost}()$
while $\tau < t$ **do**
 $\Delta \sim \exp(\sqrt{s/q})$
 $t \leftarrow \min(t, \tau + \Delta)$
 $\tau \leftarrow \text{othersNextPost}()$
end
return t

To find the optimal control $u(t, t_f]$ and cost-to-go J , we break the problem into smaller subproblems, using the Bellman's principle of optimality (proven in the Appendix):

LEMMA 3 (BELLMAN'S PRINCIPLE OF OPTIMALITY). *The optimal cost satisfies the following recursive equation:*

$$\begin{aligned} J(r(t), \lambda(t), t) &= \min_{u(t, t+dt]} \{ \mathbb{E} [J(r(t+dt), \lambda(t+dt), t+dt)] \\ & \quad + \ell(r(t), u(t)) dt \}. \end{aligned} \quad (8)$$

where the expectation is taken over all trajectories of the control and noise jump processes, N and M , in $(t, t+dt]$.

Then, we use the Bellman's principle of optimality to derive a partial differential equation on J , often called HJB equation [13]. To do so, we first assume J is continuous and then rewrite Eq. 8 as

$$\begin{aligned} J(r(t), \lambda(t), t) &= \min_{u(t, t+dt]} \{ \mathbb{E} [J(r(t), \lambda(t), t) + dJ(r(t), \lambda(t), t)] \\ & \quad + \ell(r(t), u(t)) dt \} \\ 0 &= \min_{u(t, t+dt]} \{ \mathbb{E} [dJ(r(t), \lambda(t), t)] + \ell(r(t), u(t)) dt \}. \end{aligned} \quad (9)$$

Then, we differentiate J with respect to time t , $r(t)$ and $\lambda(t)$ using the following lemma (proven in the Appendix).

LEMMA 4. *Let $x(t)$ and $y(t)$ be two jump-diffusion processes defined by following jump SDEs:*

$$\begin{aligned} dx(t) &= f(x(t), t)dt + h(x(t), t)dN(t) + g(x(t), t)dM(t) \\ dy(t) &= m(y(t), t)dt + n(y(t), t)dM(t), \end{aligned}$$

where $N(t)$, $M(t)$ are independent jump processes. If function $F(x, y, t)$ is once continuously differentiable in x , y and t , then,

$$\begin{aligned} dF(x(t), y(t), t) &= (F_t + fF_x + mF_y)(x(t), y(t), t)dt \\ & \quad + [F(x(t) + h(x(t), t), y(t), t) - F(x(t), y(t), t)] dN(t) \\ & \quad + [F(x(t) + g(x(t), t), y(t) + n(y(t), t), t) - \\ & \quad \quad F(x(t), y(t), t)] dM(t). \end{aligned}$$

Specifically, consider $x(t) = r(t)$, $y(t) = \lambda(t)$ and $F = J$ in the above lemma, then,

$$\begin{aligned} dJ(r(t), \lambda(t), t) &= J_t(r(t), \lambda(t), t)dt \\ & \quad + [\lambda'_0(t) + w\lambda_0(t) - w\lambda(t)] J_\lambda(r(t), \lambda(t), t)dt \\ & \quad + [J(0, \lambda(t), t) - J(r(t), \lambda(t), t)] dN(t) \\ & \quad + [J(r(t) + 1, \lambda(t) + \alpha, t) - J(r(t), \lambda(t), t)] dM(t) \end{aligned}$$

Next, if we plug the above equation in Eq. 9, it follows

Algorithm 2: Optimal posting times with an oracle.

Input: Initial state r_0 , interval widths w_1, \dots, w_{m+1} , parameter q and significance $s(t) = s$
Output: Overall cost $J(r_0, 0)$, optimal control u_0^*, \dots, u_m^*
for $r \leftarrow r_0 + m$ **to** 0 **do**
 $J(r, m+1) \leftarrow \frac{1}{2}r^2$
end
for $k \leftarrow m$ **to** 0 **do**
 for $r \leftarrow r_0 + k - 1$ **to** 0 **do**
 $J(r, k) = \min\{\frac{1}{2}q + J(0, k+1), \frac{1}{2}sw_{k+1}(r+1)^2 + J(r+1, k+1)\}$
 end
end
for $k \leftarrow 0$ **to** m **do**
 if $\frac{1}{2}q + J(0, k+1) < \frac{1}{2}sw_{k+1}(r_k + 1)^2 + J(r_k + 1, k+1)$ **then**
 $u_k^* \leftarrow 1; r_{k+1} \leftarrow 0$
 else
 $u_k^* \leftarrow 0; r_{k+1} \leftarrow r_k + 1$
 end
end
return $J(r_0, 0), u_0^*, \dots, u_m^*$

that

$$\begin{aligned} 0 = & \min_{u(t, t+dt)} \left\{ J_t(r(t), \lambda(t), t) dt \right. \\ & + [\lambda'_0(t) + w\lambda_0(t) - w\lambda(t)] J_\lambda(r(t), \lambda(t), t) dt \\ & + [J(0, \lambda(t), t) - J(r(t), \lambda(t), t)] \mathbb{E}[dN(t)] \\ & + [J(r(t) + 1, \lambda(t) + \alpha, t) - J(r(t), \lambda(t), t)] \mathbb{E}[dM(t)] \\ & \left. + \ell(r(t), u(t)) dt \right\}. \end{aligned} \quad (10)$$

Now, using $\mathbb{E}[dN(t)] = u(t)dt$ and $\mathbb{E}[dM(t)] = \lambda(t)dt$, and rearranging terms, the HJB equation follows:

$$\begin{aligned} 0 = & J_t(r(t), \lambda(t), t) + [\lambda'_0(t) + w\lambda_0(t) - w\lambda(t)] J_\lambda(r(t), \lambda(t), t) \\ & + [J(r(t) + 1, \lambda(t) + \alpha, t) - J(r(t), \lambda(t), t)] \lambda(t) \\ & + \min_{u(t, t+dt)} \left\{ \ell(r(t), u(t)) + [J(0, \lambda(t), t) - J(r(t), \lambda(t), t)] u(t) \right\}. \end{aligned} \quad (11)$$

To be able to continue further, we need to define the loss ℓ and the penalty ϕ functions. Following the literature on stochastic optimal control [13], we consider the following quadratic forms, which will turn out to be a tractable choice⁴:

$$\phi(r(t_f)) = \frac{1}{2}r^2(t_f) \text{ and } \ell(r(t), u(t)) = \frac{1}{2}s(t)r^2(t) + \frac{1}{2}qu^2(t),$$

where $s(t)$ is a time significance function $s(t) \geq 0$, which favors some periods of times (*e.g.*, times in which the follower is online⁵), and q is a given parameter, which trade-offs the visibility and the number of broadcasted posts.

Under these definitions, we take the derivative with respect to $u(t)$ of Eq. 11 and uncover the relationship between the optimal intensity and the optimal cost:

$$u^*(t) = q^{-1} [J(r(t), \lambda(t), t) - J(0, \lambda(t), t)]. \quad (12)$$

Finally, we substitute the above expression in Eq. 11 and find that the optimal cost J needs to satisfy the following

⁴Considering other losses with a specific semantic meaning (*e.g.*, $\mathbb{1}(r(t) \leq k)$) is a challenging direction for future work.

⁵Such information may be hidden but one can use the followers' posting activity or geographic location as a proxy [17].

nonlinear differential equation:

$$\begin{aligned} 0 = & J_t(r(t), \lambda(t), t) + [\lambda'_0(t) + w\lambda_0(t) - w\lambda(t)] J_\lambda(r(t), \lambda(t), t) \\ & + [J(r(t) + 1, \lambda(t) + \alpha, t) - J(r(t), \lambda(t), t)] \lambda(t) \\ & + \frac{1}{2}s(t)r^2(t) - \frac{1}{2}q^{-1} [J(r(t), \lambda(t), t) - J(0, \lambda(t), t)]^2 \end{aligned} \quad (13)$$

with $J(r(t_f), \lambda(t_f), t_f) = \phi(r(t_f))$ as the terminal condition. The following lemma provides us with a solution to the above equation (proven in the Appendix):

LEMMA 5. Any solution to the nonlinear differential equation given by Eq. 13 can be approximated as closely as desired by

$$J(r(t), \lambda(t), t) = f(t) + \sqrt{s(t)q}r(t) + \sum_{j=1}^m g_j(t)\lambda^j(t),$$

where $f(t)$ and $g_j(t)$ are time-varying functions, and m controls for the approximation guarantee.

Given the above Lemma and Eq. 12, the optimal intensity is readily given by following theorem:

THEOREM 6. The optimal intensity for the when-to-post problem defined by Eq. 6 with quadratic loss and penalty functions is given by $u^*(t) = \sqrt{s(t)/q}r(t)$.

The optimal intensity only depends on the position of the most recent post by user i in her follower's feed and thus allows for a very efficient procedure to sample posting times, which exploits the superposition theorem [19]. The key idea is as follows: at any given time t , we can view the process defined by the optimal intensity as a superposition of $r(t)$ inhomogeneous poisson processes with intensity $\sqrt{s(t)/q}r(t)$ which starts at jumps of the rank $r(t)$, and find the next sample by computing the minimum across all samples from these processes. Algorithm 1 summarizes our (sampling) method, which we name REDQUEEN [4]. Within the algorithm, *othersNextPost()* returns the time of the next event by other broadcasters in the followers' feeds, once the event happens. In practice, we only need to know if the event happens before we post. Remarkably, it only needs to sample $M(t_f)$ times from a (exponential) distribution (if significance is constant) and requires $O(1)$ space.

Optimizing for multiple followers. Given a broadcaster i with $N_i(t) = N(t)$ and $\mu_i(t) = u(t)$ and her followers $\mathcal{N}(i)$ with $\mathbf{M}_i(t) = \mathbf{M}(t)$ and $\boldsymbol{\gamma}_i(t) = \boldsymbol{\lambda}(t)$, the dynamics of $\mathbf{M}(t)$ and $\mathbf{r}(t)$, which we need to solve Eq. 5, are given by:

$$\begin{aligned} d\mathbf{r}(t) &= -\mathbf{r}(t) dN(t) + d\mathbf{M}(t) \\ d\boldsymbol{\lambda}(t) &= [\boldsymbol{\lambda}'_0(t) + \mathbf{w} \odot \boldsymbol{\lambda}_0(t) - \mathbf{w} \odot \boldsymbol{\lambda}(t)] dt + \boldsymbol{\alpha} \odot d\mathbf{M}(t), \end{aligned}$$

where \odot is the element-wise product and $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]^T$ and $\mathbf{w} = [w_1, \dots, w_n]^T$ are the parameters defining each of the followers' feed dynamics, and $n = |\mathcal{N}(i)|$ is the number of followers.

Consider the following quadratic forms for the loss ℓ and the penalty ϕ :

$$\begin{aligned} \phi(\mathbf{r}(t_f)) &= \sum_{i=1}^n \frac{1}{2}r_i^2(t_f) \\ \ell(\mathbf{r}(t), \mathbf{u}(t), t) &= \sum_{i=1}^n \frac{1}{2}s_i(t)r_i^2(t) + \frac{1}{2}qu^2(t). \end{aligned}$$

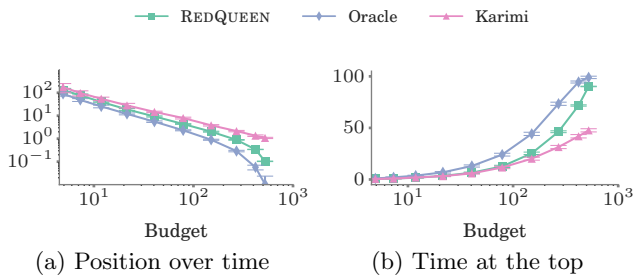


Figure 2: Optimizing for one follower. Performance of REDQUEEN in comparison with the oracle and the method by Karimi et al. [17] against number of broadcasted events. The feeds counting processes $M(t)$ due to other broadcasters are Hawkes processes with $\lambda_0 = 10$, $\alpha = 1$ and $w = 10$. In all cases, the time horizon $t_f - t_0$ is chosen such that the number of stories posted by other broadcasters is ~ 1000 . Error bars are too small to be seen.

where $s_i(t)$ is the time significance function for follower i , as defined above, and q is a given parameter. Then, proceeding similarly as in the case of one follower, we can show that the optimal intensity $u^*(t)$ is given by:

$$u^*(t) = \sum_{i=1}^n \sqrt{s_i(t)/q} r_i(t), \quad (14)$$

which only depends on the position of the most recent post by user i in her followers' feeds. Finally, we can readily adapt REDQUEEN (Algorithm 1) to efficiently sample the posting times using the above intensity – it only needs to sample $|\cup_{j \in \mathcal{N}(i)} \mathcal{F}_{j \setminus i}(t_f)|$ values and requires $O(|\mathcal{N}(i)|)$ space.

Optimizing with an oracle. In this section, we consider a broadcaster i with $N_i(t) = N(t)$ and $\mu_i(t) = u(t)$, only one of her followers j with $M_{j \setminus i}(t) = M(t)$, and a constant significance $s(t) = s$. The derivation can be easily adapted to the case of multiple followers and time-varying significance.

Suppose there is an (idealized) oracle that reveals $M(t)$ from t_0 to t_f , i.e., the history $\mathcal{F}_{j \setminus i}(t_f) = \mathcal{F}(t_f)$ is given, and $M(t_f) = |\mathcal{F}(t_f)| = m$. Then, we can rewrite Eq. 5 as

$$\begin{aligned} & \underset{u(t_0, t_f)}{\text{minimize}} \quad \mathbb{E}_{N(t_0, t_f)} \left[\phi(r(t_f)) + \int_{t_0}^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \\ & \text{subject to} \quad u(t) \geq 0 \quad \forall t \in (t_0, t_f), \end{aligned}$$

where the expectation is only taken over all possible realizations of the counting process $N(t_0, t_f)$ since $M(t_0, t_f)$ is revealed by the oracle and thus deterministic.

Similar to the previous sections, assume the loss ℓ and penalty ϕ are quadratic. It is easy to realize that the best times for user i to post will always coincide with one of the times in $\mathcal{F}(t_f)$. More specifically, given a posting time $\tau_i \in (t_k, t_{k+1})$, where $t_k, t_{k+1} \in \mathcal{F}(t_f)$, one can reduce the cost by $(1/2)q(\tau_i - t_k)r^2(t_k)$ by choosing instead to post at t_k . As a consequence, we can discretize the dynamics of $r(t)$ in times $\mathcal{F}(t_f)$, and write $r_{k+1} = r_k + 1 - (r_k + 1)u_k$, where $r_k = r(t_k^-)$, $u_k = u(t_k^+) \in \{0, 1\}$, $t_k \in \mathcal{F}(t_f)$. We can easily see that r_k is bounded by $0 \leq r_k < r_0 + m$. Similarly, we

can derive the optimal cost-to-go in discrete-time as:

$$J(r_k, k) = \min_{u_k, \dots, u_m} \frac{1}{2} r_{m+1}^2 + \sum_{i=k}^m \frac{1}{2} q w_{i+1} r_{i+1}^2 + \frac{1}{2} s u_i^2,$$

where $w_i = t_i - t_{i-1}$. Next, we can break the minimization and use Bellman's principle of optimality,

$$J(r_k, k) = \min_{u_k} \frac{1}{2} q w_{k+1} r_{k+1}^2 + \frac{1}{2} s u_k^2 + J(r_{k+1}, k+1),$$

and, since $u_k \in \{0, 1\}$, the above recursive equation can be written as

$$J(r_k, k) = \min \left\{ \frac{1}{2} s + J(0, k+1), \frac{1}{2} q w_{k+1} (r_k + 1)^2 + J(r_k + 1, k+1) \right\}.$$

Finally, we can find the optimal control u_k^* , $k = 0, \dots, m$ and cost $J(r_0, 0)$ by backtracking from the terminal condition $J(r_{m+1}, m+1) = r_{m+1}^2/2$ to the initial state r_0 , as summarized in Algorithm 2, which can be adapted to multiple followers. Note that, in this case, the optimal strategy is not stochastic and consists of a set of optimal posting times, as one could have guessed. However, for multiple followers, the complexity of the algorithm is $O(m^2)$, where $m = |\cup_{j \in \mathcal{N}(i)} \mathcal{F}_{j \setminus i}(t_f)|$.

5. EXPERIMENTS

5.1 Experiments on synthetic data

Experimental setup. We evaluate the performance via two quality measures: position over time, $\int_0^T r(t) dt$, and time at the top, $\int_0^T \mathbb{1}(r(t) < 1) dt$ and compare the performance of REDQUEEN against the oracle, described in Section 4, and the method by Karimi et al. [17], which, to the best of our knowledge, is the state of the art. Unless otherwise stated, we set the significance $s_i(t) = 1, \forall t, i$ and use the parameter q to control the number of posts by REDQUEEN⁶.

Optimizing for one follower. We first experiment with one broadcaster and one follower against an increasing number of events (or budget). We generate the counting processes $M(t)$ due to other broadcasters using Hawkes processes, which are particular instances of the general functional form given by Eq. 1. We perform 10 independent simulation runs and compute the average and standard error (or standard deviation) of the quality measures. Figure 2 summarizes the results, which show that our method: (i) consistently outperforms the method by Karimi et al. by large margins; (ii) achieves at most $3\times$ higher position over time than the oracle as long as the budget is $< 30\%$ of the posted events by all other broadcasters; and, (iii) achieves $> 40\%$ of the value of time at the top that the oracle achieves.

Optimizing for multiple followers. Next, we experiment with one broadcaster and multiple followers. In this case, we generate the counting processes $M(t)$ due to other broadcasters using piece-constant intensity functions. More specifically, we simulate the feeds of each follower for 1 day,

⁶The expected number of posts by REDQUEEN are a decreasing function of q . Hence, we can use binary search to guess q and then use averaging over multiple simulation runs to estimate the number of posts made.

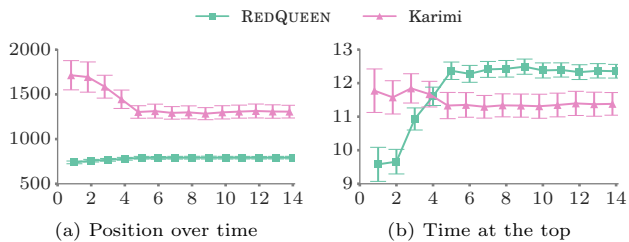


Figure 3: Optimizing for multiple followers. Performance of REDQUEEN in comparison with the method by Karimi et al. [17] against number of followers. The feeds counting processes $M(t)$ due to other broadcasters follow piecewise constant intensities, where the intensity of each follower remains constant within each piece, it varies as a half-sinusoid across pieces and it starts with a random initial phase. The performance of both methods stays constant upon addition of more followers.

using 24 1-hour long segments, where the rate of posts remains constant per follower in each segment and the rate itself varies as a half-sinusoid (*i.e.*, from $\sin 0$ to $\sin \pi$), with each follower starting with a random initial phase. This experimental setup reproduces volume changes throughout the day across followers’ feeds in different time-zones and closely resembles the settings in previous work [17]. The total number of posts by the REDQUEEN broadcaster is kept nearly constant and is used as the budget for the other baselines. Additionally, for Karimi’s method, we provide as input the true empirical rate of tweets per hour for each user. Here, we do not compare with the oracle since, due to its quadratic complexity, it does not scale.

Figure 3 summarizes the results. In terms of position over time, REDQUEEN outperforms Karimi’s method by a factor of 2. In terms of time at the top, REDQUEEN achieves $\sim 18\%$ lower values than Karimi’s method for 1-4 followers but $\sim 10\%$ higher values for >5 followers. A potential reason for Karimi’s method to perform best in terms of time at the top for a low number of followers and piecewise constant intensities is that, while the number of followers is low, there are segments which are clearly favorable and thus Karimi’s method concentrates posts on those, however, as the number of followers increases, there are no clear favorable segments and thus advance planning does not give Karimi’s method any advantage. On the other hand, REDQUEEN, due to its online nature, is able to adapt to transient variations in the feeds.

5.2 Experiments on real data

Dataset description and experimental setup. We use data gathered from Twitter as reported in previous work [5], which comprises profiles of 52 million users, 1.9 billion directed follow links among these users, and 1.7 billion public tweets posted by the collected users. The follow link information is based on a snapshot taken at the time of data collection, in September 2009. Here, we focus on the tweets published during a two month period, from July 1, 2009 to September 1, 2009, in order to be able to consider the social graph to be approximately static, and sample 2000 users uniformly at random as broadcasters and record all the tweets they posted. Then, for each of these broadcasters, we

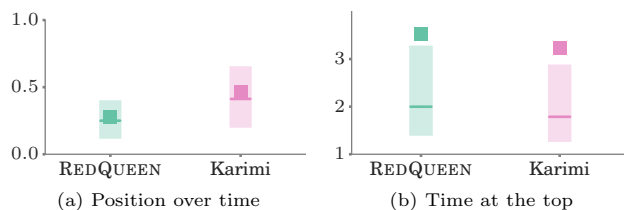


Figure 4: Performance of REDQUEEN and the method by Karimi et al. [17] for 2000 Twitter users, picked at random. The solid horizontal line (square) shows the median (mean) quality measure, normalized with respect to the value achieved by the users’ actual *true* posts, and the box limits correspond to the 25%-75% percentiles.

track down their followers and record all the (re)tweets they posted as well as reconstruct their timelines by collecting all the (re)tweets published by the people they follow. We assign equal significance to each follower but filter out those who follow more than 500 people since, otherwise, they would dominate the optimal strategy. Finally, we tune q such that the total number of tweets posted by our method is equal to the number of tweets the broadcasters tweeted during the two month period (with a tolerance of 10%).

Solution quality. We only compare the performance of our method against the method by Karimi et al. [17] since the oracle does not scale to the size of real data. Moreover, for the method by Karimi et al., we divide the two month period into ten segments of approximately one week to fit the piecewise constant intensities of the followers’ timelines, which the method requires.

Figure 4 summarizes the results by means of box plots, where position over time and time at the top are normalized with respect to the value achieved by the broadcasters’ actual *true* posts during the two month period. That means, if $y = 1$, the optimized intensity achieves the same position over time or time at the top as the broadcaster’s true posts. In terms of position over time and time at the top, REDQUEEN consistently outperforms competing methods by large margins and achieves $0.28\times$ lower average position and $3.5\times$ higher time at the top, in average, than the broadcasters’ true posts – in fact, it achieves lower position over time (higher time at the top) for 100% (99.1%) of the users.

Time significance. We look at the actual broadcasting strategies for one real user and investigate the effect of a time varying significance. We define $s_i(t)$ to be the probability that follower i is online on that weekday, estimated empirically using the (re)tweets the follower posted as in Karimi et al. [17]. Figure 5 compares the position over time for the most recent tweet posted by a real user against the most recent one *posted* by a simulation run of REDQUEEN with and without time varying significance. We can see that without significance information, REDQUEEN posts at nearly an even pace. However, when we supply empirically estimated significance, REDQUEEN avoids tweeting at times the followers are unlikely to be active, *i.e.*, the weekends, denoted by the shaded areas in panel (c) of Figure 5. Due to this, the average position (maximum position) falls from 389.45 (1085.17) to 425.25 (1431.0), but is still lower than 698.04 (2597.9) obtained by the user’s original posting schedule.

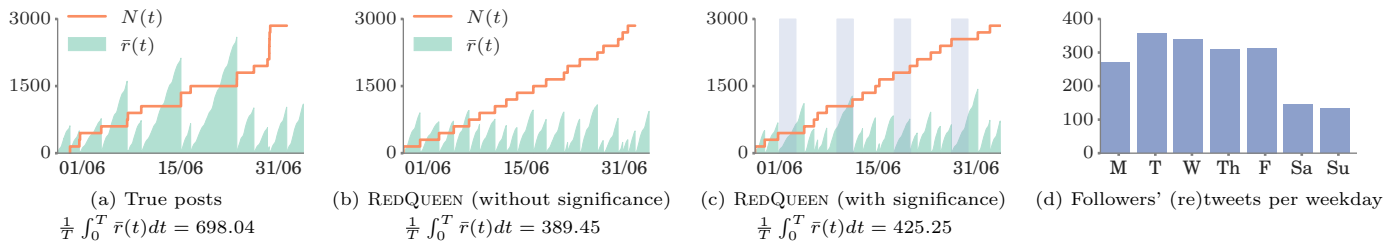


Figure 5: A broadcaster chosen from real data. Panels compares the position over time $\bar{r}(t) = \sum_{i=0}^{\mathcal{N}} r(t)/\mathcal{N}$ (in green; lower is better) for the most recent tweet posted by a real user against the most recent one *posted* by a simulation run of REDQUEEN without and with significance. Here, the orange staircases represent the counts $N(t)$ of the tweets posted by the real user and REDQUEEN over time. The shaded area in panel (c) highlights weekends. We can see that REDQUEEN avoided tweeting on weekends, when the followers are less likely to be active/logged-in, as seen in panel (d).

6. CONCLUSIONS

In this paper, we approached the when-to-post problem from the perspective of stochastic optimal control and showed that the optimal broadcasting strategy is surprisingly simple – it is given by the position of her most recent post on each of her follower’s feed. Such a strategy can be implemented using a simple and efficient on-line algorithm. We experimented with synthetic and real-world data gathered from Twitter and showed that our algorithm consistently makes a user’s posts more visible over time and it significantly outperforms the state of the art.

Our work also opens many venues for future work. For example, in this work, we considered social networks that sort stories in the users’ feeds in inverse chronological order (*e.g.*, Twitter, Weibo). Extending our methodology to social networks that sort stories algorithmically (*e.g.*, Facebook) is a natural next step. Currently, REDQUEEN optimizes a quadratic loss on the position of a broadcaster’s most recent post on her followers’ feeds over time. However, it would be useful to derive optimal broadcasting intensities for other losses, *e.g.*, time at the top. Moreover, we assume that only one broadcaster is using REDQUEEN. A very interesting follow-up would be augmenting our framework to consider multiple broadcasters under cooperative, competitive and adversarial environments. Finally, the novel technical aspects of our problem formulation, *e.g.*, optimal control of jump SDEs with double stochastic temporal point processes, can be applied to other control problems in social and information networks such as activity shaping [10] and opinion control [27].

7. REFERENCES

- [1] O. Aalen, O. Borgan, and H. K. Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.
- [2] L. Backstrom, E. Bakshy, J. M. Kleinberg, T. M. Lento, and I. Rosenn. Center of attention: How facebook users allocate attention across friends. *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, 2011.
- [3] R. Blumenthal. An extended markov property. *Transactions of the American Mathematical Society*, 85(1):52–72, 1957.
- [4] L. Carroll. *Through the looking glass: And what Alice found there*. Rand, McNally, 1917.
- [5] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010.
- [6] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2010.
- [7] J. Cheng, L. Adamic, P. Dow, J. Kleinberg, and J. Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World Wide Web*, 2014.
- [8] T. Chenhao, L. Lee, and B. Pang. The effect of wording on message propagation: Topic-and author-controlled natural experiments on twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.
- [9] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in Neural Information Processing Systems*, 2013.
- [10] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. In *Advances in Neural Information Processing Systems*, 2014.
- [11] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, 2015.
- [12] M. Gomez-Rodriguez, K. Gummadi, and B. Schölkopf. Quantifying information overload in social media and its impact on social contagions. In *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, 2014.
- [13] F. B. Hanson. *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*. SIAM, 2007.
- [14] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.
- [15] N. Hodas and K. Lerman. How visibility and divided attention constrain social contagion. *SocialCom*, 2012.
- [16] J.-H. Kang and K. Lerman. Vip: Incorporating human cognitive biases in a probabilistic model of retweeting.

In *Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 101–110. Springer, 2015.

- [17] M. R. Karimi, E. Tavakoli, M. Farajtabar, L. Song, and M. Gomez-Rodriguez. Smart broadcasting: Do you want to be seen? In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [18] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2003.
- [19] J. F. C. Kingman. *Poisson processes*. Oxford university press, 1992.
- [20] H. Lakkaraju, J. J. McAuley, and J. Leskovec. What’s in a name? understanding the interplay between titles, content, and communities in social media. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, 2013.
- [21] K. Lerman and T. Hogg. Leveraging position bias to improve peer recommendation. *PloS one*, 9(6), 2014.
- [22] G. Miritello, R. Lara, M. Cebrian, and E. Moro. Limited communication capacity unveils strategies for human interaction. *Scientific reports*, 3, 2013.
- [23] N. Navaroli and P. Smyth. Modeling response time in digital human communication. In *Proceedings of the 9th International AAAI Conference on Web and Social Media*, 2015.
- [24] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2002.
- [25] N. Spasojevic, Z. Li, A. Rao, and P. Bhattacharyya. When-to-post on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2127–2136. ACM, 2015.
- [26] M. H. Stone. The generalized weierstrass approximation theorem. *Mathematics Magazine*, 21(5):237–254, 1948.
- [27] Y. Wang, E. Theodorou, A. Verma, and L. Song. Steering opinion dynamics in information diffusion networks. *arXiv preprint arXiv:1603.09021*, 2016.
- [28] A. Zarezade, A. Khodadadi, M. Farajtabar, H. R. Rabiee, and H. Zha. Correlated Cascades: Compete or Cooperate. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2016.
- [29] Q. Zhao, M. Erdogdu, H. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

APPENDIX

Proof of Proposition 1

Using the left continuity of poisson processes, we have that:

$$d\lambda^*(t) = \lambda'_0(t)dt + \alpha d\left(\int_0^t g(t-s)dN(s)\right).$$

Then, using Ito’s calculus [13], we can rewrite the differential in the second term as

$$\begin{aligned} & \int_0^{t+dt} g(t+dt-s) dN(s) - \int_0^t g(t-s) dN(s) \\ &= \int_0^{t+dt} (g(t-s) + g'(t-s)dt) dN(s) - \int_0^t g(t-s) dN(s) \\ &= \int_t^{t+dt} g(t-s) dN(s) + dt \int_0^{t+dt} g'(t-s) dN(s) \\ &= g(0)dN(t) - w dt \int_0^{t+dt} g(t-s) dN(s) \\ &= dN(t) - w dt \int_0^t g(t-s) dN(s) \\ &= dN(t) + \frac{w}{\alpha}[\lambda_0(t) - \lambda^*(t)]dt. \end{aligned}$$

This completes the proof.

Proof of Lemma 3

$$\begin{aligned} & \min_{u(t,t_f)} \mathbb{E}_{(N,M)(t,t_f)} \left[\phi(r(t_f)) + \int_t^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \\ &= \min_{u(t,t_f)} \mathbb{E}_{(N,M)(t,t_f)} \left[\phi(r(t_f)) + \int_t^{t+dt} \ell(r(\tau), u(\tau)) d\tau \right. \\ & \quad \left. + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \\ &= \min_{u(t,t_f)} \mathbb{E}_{(N,M)(t,t+dt)} \left[\mathbb{E}_{(N,M)(t+dt,t_f)} \left[\phi(r(t_f)) + \ell(t, r, u) dt \right. \right. \\ & \quad \left. \left. + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \right] \\ &= \min_{u(t,t+dt)} \min_{u(t+dt,t_f)} \mathbb{E}_{(N,M)(t,t+dt)} \left[\ell(r(t), \lambda(t), t) dt \right. \\ & \quad \left. + \mathbb{E}_{(N,M)(t+dt,t_f)} \left[\phi(r(t_f)) + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau)) d\tau \right] \right] \\ &= \min_{u(t,t+dt)} \mathbb{E}_{(N,M)(t,t+dt)} [J(\lambda(t+dt), r(t+dt), t+dt)] \\ & \quad + \ell(r(t), u(t)) dt. \end{aligned}$$

Proof of Lemma 4

According to the definition of differential,

$$\begin{aligned} dF &:= dF(x(t), y(t), t) \\ &= F(x(t+dt), y(t+dt), t+dt) - F(x(t), y(t), t) \\ &= F(x(t) + dx(t), y(t) + dy(t), t+dt) - F(x(t), y(t), t) \end{aligned}$$

where we used the complete notation for F to be more clear. Using the zero-one jump law of point processes [19], we can write:

$$\begin{aligned} dF &= F(x + fdt + h, y + mdt, t + dt) dN(t) \\ & \quad + F(x + fdt + g, y + mdt + n, t + dt) dM(t) \\ & \quad + F(x + fdt, y + mdt, t + dt) (1 - dN(t))(1 - dM(t)) \\ & \quad - F(x, y, t) \end{aligned}$$

where for notational simplicity we drop arguments of all functions except F . Then, we can expand the first three

terms in the right hand sides:

$$\begin{aligned}
F(x + fdt + h, y + mdt, t + dt) &= F(x + h, y, t) \\
+ F_x(x + h, y, t)fdt + F_y(x + h, y, t)mdt + F_t(x + h, y, t)dt \\
F(x + fdt + g, y + mdt + n, t + dt) &= F(x + g, y + n, t) \\
+ F_x(x + g, y + n, t)fdt + F_y(x + g, y + n, t)mdt + F_t(x + g, y + n, t)dt \\
F(x + fdt, y + mdt, t + dt) &= F(x, y, t) + F_x(x, y, t)fdt \\
+ F_y(x, y, t)mdt + F_t(x, y, t)dt,
\end{aligned}$$

using that the bilinear differential form $dt dN(t) = 0$ [13] and $dN(t)dM(t) = 0$ by the zero-one jump law. Finally,

$$\begin{aligned}
dF &= (fF_x + mF_y + F_t)(x, y, t)dt + [F(x + h, y, t) - \\
&F(x, y, t)]dN(t) + [F(x + g, y + n, t) - F(x, y, t)]dM(t).
\end{aligned}$$

Proof of Lemma 5

According to the Stone-Weierstrass theorem, any continuous function in a closed interval can be approximated as closely as desired by a polynomial function [26]. So by assuming the continuity of cost function we consider the general form,

$$J(r(t), \lambda(t), t) = \sum_{i=0}^n \sum_{j=0}^m f_{ij}(t) r^i(t) \lambda^j(t),$$

where m and n are arbitrary large numbers. Indeed in each time t we approximate a two variate function of $r(t)$ and $\lambda(t)$ by a polynomial where the coefficient are defined by the time varying functions $f_{ij}(t)$. If we substitute this function in to Eq. 13 and simplifying the expression we would have,

$$\begin{aligned}
0 &= \sum_{i=1}^n f'_{i0} r^i + f_{i0} (r + 1)^i \lambda - f_{i0} r^i \lambda \\
&+ \sum_{j=1}^m f'_{0j} \lambda^j + j(\lambda'_0 + w\lambda_0 - w\lambda) f_{0j} \lambda^{j-1} + f_{0j} (\lambda + \alpha)^j \lambda - f_{0j} \lambda^{j+1} \\
&+ \sum_{i=1}^n \sum_{j=1}^m j(\lambda'_0 + w\lambda_0 - w\lambda) f_{ij} r^i \lambda^{j-1} \\
&+ \sum_{i=1}^n \sum_{j=1}^m f_{ij} (r + 1)^i (\lambda + \alpha)^j \lambda - f_{ij} r^i \lambda^{j+1} \\
&- \frac{1}{2} q^{-1} \left[\sum_{i=1}^n f_{i0} r^i + \sum_{i=1}^n \sum_{j=1}^m f_{ij} r^i \lambda^j \right]^2 + \frac{1}{2} s r^2 + f'_{00}
\end{aligned}$$

where for notational simplicity we omitted the time argument of functions. To find the unknown functions $f_{ij}(t)$, we equate the coefficient of different variables. If we consider the coefficient of r^{2n} , we have $f_{n0}(t) = 0$. We can continue this argument for $n - 1, n - 2, \dots, 2$ to show that $\forall i \geq 2; f_{i0}(t) = 0$. Similar reasoning for coefficients of $r^{2i} \lambda^{2j}$ shows that $\forall j, i \geq 2; f_{ij}(t) = 0$. Finally, the coefficient of r^2 is $s(t)/2 - q^{-1} f_{10}^2(t)/2 = 0$, so $f_{10}(t) = \sqrt{s(t)q}$. If we rename $f_{0j}(t)$ to $g_j(t)$ and $f_{00}(t)$ to $f(t)$, then we have

$$J(r(t), \lambda(t), t) = f(t) + \sqrt{s(t)q} r(t) + \sum_{j=1}^m g_j(t) \lambda^j(t).$$

We can continue the previous method to find the remaining coefficients and completely define the cost-to-go function. If we equate the coefficient of λ^j to zero we would have a system of first order differential equations which its j 'th row

for $j > 1$ is:

$$\begin{aligned}
g'_j(t) + j(\alpha - w)g_j(t) + (j + 1)(\lambda'_0(t) + w\lambda_0(t) + \frac{j}{2}\alpha^2)g_{j+1}(t) \\
- g_{j-1}(t) + \sum_{k=2}^{m-j} \binom{j+k}{k+1} \alpha^{k+1} g_{j+k}(t) = 0.
\end{aligned}$$

We can also evaluate the corresponding differential equation for $j = 1$. When $\lambda_0(t) = \lambda_0$, we can express these differential equations using a matrix differential equation $\mathbf{g}'(t) = A\mathbf{g}(t)$. and its solution is $\mathbf{g}(t) = c_1 e^{\zeta_1 t} \mathbf{u}_1 + c_2 e^{\zeta_2 t} \mathbf{u}_2 + \dots + c_n e^{\zeta_n t} \mathbf{u}_n$ where ζ_i and \mathbf{u}_i are eigenvalue and eigenvector of matrix A and c_i is a constant found using the terminal conditions. Since in triangular matrices diagonal entries are eigenvalues, we have $\mathbf{g}(t) = \sum_{j=1}^m c_j e^{j(w-\alpha)t} \mathbf{u}_j$. We can approximate general time varying $\lambda_0(t)$ using a piece-wise function and repeat the above procedure for each piece.