Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation

Jooyeon Kim
KAIST
jooyeon.kim@kaist.ac.kr

Behzad Tabibian
MPI for Intelligent Systems
MPI for Software Systems
me@btabibian.com

Alice Oh
KAIST
alice.oh@kaist.edu

Bernhard Schölkopf
MPI for Intelligent Systems
bs@tuebingen.mpg.de

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

ABSTRACT

Online social networking sites are experimenting with the following *crowd-powered* procedure to reduce the spread of fake news and misinformation: whenever a user is exposed to a story through her feed, she can flag the story as misinformation and, if the story receives *enough* flags, it is sent to a trusted third party for fact checking. If this party identifies the story as misinformation, it is marked as *disputed*. However, given the uncertain number of exposures, the high cost of fact checking, and the trade-off between flags and exposures, the above mentioned procedure requires careful reasoning and smart algorithms which, to the best of our knowledge, do not exist to date.

In this paper, we first introduce a flexible representation of the above procedure using the framework of marked temporal point processes. Then, we develop a scalable online algorithm, Curb, to select *which* stories to send for fact checking and *when* to do so to efficiently reduce the spread of misinformation with provable guarantees. In doing so, we need to solve a novel stochastic optimal control problem for stochastic differential equations with jumps, which is of independent interest. Experiments on two real-world datasets gathered from Twitter and Weibo show that our algorithm may be able to effectively reduce the spread of fake news and misinformation.

1 INTRODUCTION

In recent years, social media and online social networking sites have become a major disseminator of false facts, urban legends, fake news, or, more generally, misinformation. In this context, there are growing concerns that misinformation on these platforms has fueled the emergence of a *post-truth* society, where debate is perniciously framed by the repeated assertion of talking points to which factual rebuttals by the media or independent experts are ignored. For example, the post-truth label has been widely used to describe the presidential campaign of Donald Trump in the 2016 U.S. elections[1] and the Leave campaign in the 2016 Brexit referendum[2], whose outcomes have been then called into question.

In an effort to curb the spread of misinformation, major online social networking sites, such as Facebook, Twitter or Weibo, are (considering) resorting to the crowd[3,4,5]. In particular, they are experimenting with the following procedure to reduce the spread of misinformation: whenever users are exposed to a story through their feeds, they have a choice to flag the story as misinformation and, if the story receives *enough* flags, it is directed to a coalition of independent organizations[6], signatories of Poynter's International Fact Checking Code of Principles[7], for fact checking. If the fact checking organizations identify a story as misinformation, it gets flagged as disputed and may also appear lower in the users' feeds, reducing the number of people who are *exposed* to misinformation. In this context, online social networking sites are giving advice to its millions of users on how to spot misinformation online[8]. However, the above mentioned procedure requires careful reasoning and smart algorithms which, to the best of our knowledge, are nonexistent to date:

— *Uncertain number of exposures:* the spread of information over social networking sites is a stochastic process, which may depend on, *e.g.*, the information content, the users' influence and the network

---

[1] https://www.washingtonpost.com/posteverything/wp/2016/06/16/why-the-post-truth-political-era-might-be-around-for-a-while/
[2] https://www.theguardian.com/commentisfree/2016/may/13/boris-johnson-donald-trump-post-truth-politician
[3] https://newsroom.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/
[4] https://www.washingtonpost.com/news/the-switch/wp/2017/06/29/twitter-is-looking-for-ways-to-let-users-flag-fake-news/
[5] http://www.scmp.com/news/china/policies-politics/article/2055179/how-chinas-highly-censored-wechat-and-weibo-fight-fake
[6] This coalition of organizations includes, among many others, Snopes (http://www.snopes.com), FactCheck (http://www.factcheck.org) or Politifact (http://www.politifact.com).
[7] http://www.poynter.org/fact-checkers-code-of-principles/
[8] https://www.nytimes.com/2017/05/08/technology/uk-election-facebook-fake-news.html

structure. Thus, the number of users exposed to different stories varies greatly and we need to consider probabilistic exposure models to capture this uncertainty.

— *Fact checking is costly:* given the myriad of (fake) stories spreading in online social networking sites and the observation that fact checking is a costly process, we can only expect (the reviewers from) the coalition of independent organizations to fact check a small percentage of the set of stories spreading over time. Therefore, it is necessary to decide *which* stories to fact check and *when* to do so.

— *Flags vs exposures:* the more users are exposed to a story before sending it for fact checking, the greater the confidence a story may be misinformation, however, the higher the potential damage if it turns out to be misinformation. Thus, we need to find the optimal trade-off between misinformation evidence, by means of flagging data, and misinformation reduction, by means of preventing (unwarned) exposures to misinformation.

Our approach. To tackle the above challenges, we first introduce a novel representation of the above procedure using the framework of marked temporal point processes [1]. Then, we find *which* stories to send for fact checking by solving a novel stochastic optimal control problem for SDEs with jumps [16], which differs from the nascent literature on stochastic optimal control of social and information systems [41, 42, 47, 48] in two technical aspects:

I. The control signal is a multidimensional survival process (*i.e.*, a terminating temporal point process), which is defined by means of a set of conditional intensities (*i.e.*, stories to fact check), while previous work has considered nonterminating temporal point processes as control signals.

II. The SDEs with jumps depend on random variables (*i.e.*, flags) whose distributions depend on *a priori* unknown parameters (*i.e.*, flagging probability, which depends on whether a story is fake or legitimate). In this context, a posterior estimate of these parameters arises naturally in the formulation of the optimal control problem. In contrast, previous work did not consider application scenarios where posterior inference was required.

These technical aspects have implications beyond the problem of detecting and reducing misinformation since they establish an unexplored connection between stochastic online optimal control, survival analysis and Bayesian inference. Moreover, we find that the solution to the above optimal control problem is relatively simple and intuitive: the optimal intensity of fact checking is proportional to a posterior estimate of the rate of misinformation. As a consequence, we can develop a scalable online algorithm, Curb, to schedule *which* stories to send for fact checking to efficiently reduce misinformation. Finally, we experiment on two real-world datasets gathered from Twitter and Weibo and show that our algorithm is able to effectively reduce the spread of misinformation and fake news. To facilitate research in this area, we release an open source implementation of our algorithm at http://learning.mpi-sws.org/curb/.

Related work. The research areas most closely related to ours are on truth discovery and rumor detection. In terms of truth discovery, most previous work discovers the truth by assessing the trustworthiness of the information sources and they do so using link-based measures [4, 7, 15], information retrieval based measures [44], accuracy-based measures [9, 10, 19, 27, 45], content-based measures [2, 14, 35, 37, 38, 43, 51], graphical models [32, 46, 49, 50], or survival analysis [26, 36]. A recent line of work [22, 24, 31, 40] also considers scenarios in which the truth may change over time. In terms of rumor detection, previous work has typically relied on textual, network and temporal features to characterize and detect rumors [13, 20, 25, 28, 29, 33, 34]. However, our work differs from previous work in several key aspects: (i) the *crowd-powered* fact checking procedure is, to the best of our knowledge, novel; (ii) we both detect and reduce the spread of misinformation, in contrast, previous work has focused on detection and has not designed interventional strategies considering the exposure dynamics; and, (iii) we develop a principled online algorithm by solving a stochastic optimal control of SDEs with jumps, which is able to adapt to changes in the rate of exposures and flags, while previous work has developed off-line algorithms. Moreover, note that our methodology could be extended to leverage the above detection methods to further refine the posterior estimate of the rate of misinformation.

Learning from the crowd has been widely used in many different areas, *e.g.*, web security [6, 39], spam detection [5, 52], phishing detection [30] and fake online account detection [12]. Moreover, using expert knowledge, such as the one provided by fact checking organizations, has been used to improve the quality and reliability of crowd learning procedures [17, 23]. However, to the best of our knowledge, the present work is the first that leverages both the crowd and expert knowledge in the context of detecting and preventing the spread of fake news and misinformation.

Finally, there has been a paucity of work on stochastic optimal control of SDEs with jumps [41, 42, 47, 48], however, previous work has focused on steering social activity, has not considered survival processes as control signals, and has not considered SDEs with jumps depending on random variables whose distributions depend on unknown parameters.

2 PROBLEM FORMULATION

To reduce the spread of misinformation, we optimize the following fact checking procedure used by major online social networking sites. Users can flag any story in their feed as misinformation and, if a story receives *enough* flags, it is sent to a third party for fact checking. If the third-party identifies a story as misinformation, it gets flagged as disputed and may also appear lower in the users' feeds. In this procedure, since the third-party fact checking is costly, we need to decide *which* stories to fact check and *when* to do so—decide how many flags are *enough*. For ease of exposition, we assume that, if a story is sent for fact checking, the story is instantaneously verified—it is instantly revealed whether the story is fake or genuine[9].

In this section, we first leverage the framework of marked temporal point processes [1] to model the above mentioned fact checking procedure, starting from the data representation the model uses,

---

[9]One could easily relax this assumption by modeling the delay between the time when a story is sent for fact checking and the time when it is verified.

then define and estimate the rate of misinformation, which we will use to decide what and when to fact check, and finally state the fact checking scheduling problem.

Data representation. Given an online social networking site with a set of users $\mathcal{U}$ and a set of unverified stories $\mathcal{S}$, we define two types of user events: *exogenous* events, which correspond to the publication of stories by users on their own initiative, and *endogenous* events, which correspond to the resharing and/or flagging of stories by users who are exposed to them through their feeds, similarly as in previous work [8, 11].

Formally, we represent each exogenous event as a triplet

$$p := (\underset{\underset{\text{user}}{\uparrow}}{u}, \quad \underset{\underset{\text{time}}{\uparrow}}{t}, \quad \overset{\overset{\text{story}}{\downarrow}}{s}),$$

which means that user $u \in \mathcal{U}$ posted story $s \in \mathcal{S}$ at time t at her own initiative. Moreover, we denote the history of exogenous events for each story s up to time t by $\mathcal{H}_s^p(t) = \bigcup_{i:t_i < t} \{p_i \mid s_i = s\}$.

Similarly, we represent each endogenous event as a 5-tuple

$$e := (\underset{\underset{\text{user}}{\uparrow}}{u}, \quad \underset{\underset{\text{time}}{\uparrow}}{t}, \quad \overset{\overset{\text{story}}{\downarrow}}{s}, \quad \underset{\underset{\text{reshare}}{\uparrow}}{r}, \quad \overset{\overset{\text{flag}}{\downarrow}}{f}),$$

which means that user $u \in \mathcal{U}$ was exposed to story $s \in \mathcal{S}$ at time t and decided (not) to reshare it, $r = 1$ ($r = 0$), and/or (not) to flag it, $f = 1$ ($f = 0$). Then, we denote the history of endogenous events for story s up to time t by $\mathcal{H}_s^e(t) = \bigcup_{i:t_i < t} \{e_i \mid s_i = s\}$. Finally, we gather the history of exogenous and endogenous events for story s up to time t by $\mathcal{H}_s(t) = \mathcal{H}_s^p(t) \bigcup \mathcal{H}_s^e(t)$ and the overall history of exogenous and endogenous events up to time t by $\mathcal{H}(t) = \bigcup_{s \in \mathcal{S}} \mathcal{H}_s(t)$.

Fact checking generative process. We represent the times of the exogenous and endogenous events within the social networking site using two multidimensional counting processes, $\mathbf{N}^p(t)$ and $\mathbf{N}^e(t)$, in which the s-th dimension, $N_s^p(t)$ and $N_s^e(t)$, count the number of exogenous and endogenous events for story s up to time t. Following the literature on temporal point processes [1], we characterize these counting processes using their corresponding intensities, *i.e.*,

$$\mathbb{E}[d\mathbf{N}^p(t)|\mathcal{H}(t)] = \lambda^p(t)dt,$$
$$\mathbb{E}[d\mathbf{N}^e(t)|\mathcal{H}(t)] = \lambda^e(t)dt,$$

where $d\mathbf{N}^p(t)$ and $d\mathbf{N}^e(t)$ denote the number of exogenous and endogenous events in the window $[t, t+dt)$ and $\lambda^p(t) := [\lambda_s^p(t)]_{s \in \mathcal{S}}$ and $\lambda^e(t) := [\lambda_s^e(t)]_{s \in \mathcal{S}}$ denote the vector of intensities associated to all stories. Every time a user is exposed to a story s, the binary indicators r and f are sampled from two Bernoulli distributions, $\mathbb{P}(r = 1) = r_s$ and $\mathbb{P}(f = 1) = f_s$, where r_s and f_s are story specific parameters which may depend on many complex factors, *e.g.*, content, source.

Moreover, we represent the times when stories are sent to a trusted third-party for fact checking using a multidimensional binary counting process, $\mathbf{M}(t)$, in which the s-th dimension, $M_s(t)$, becomes one when story s is sent for fact checking. Here, we characterize this process using a vector of intensities $\mathbf{u}(t) := [u_s(t)]_{s \in \mathcal{S}}$ associated to all stories, *i.e.*, $\mathbb{E}[d\mathbf{M}(t)|\mathcal{H}(t)] = \mathbf{u}(t) \odot (1 - \mathbf{M}(t))dt$, where $\odot$ denotes pairwise product and the term $(1 - \mathbf{M}(t))$ ensures each story is sent for fact checking only once. Whenever story s is

sent for fact checking, we assume it gets instantly verified—it gets revealed whether s is fake or genuine. Under this characterization, deciding *which* stories to fact check and *when* to do so becomes a problem of finding the vector of intensities $\mathbf{u}(t)$ of fact checking events.

Endogenous and exogenous intensities. We model exogenous events for each unverified story $s \in \mathcal{S}$ as:

$$\lambda_s^p(t) := h_s(t)(1 - M_s(t)), \quad (1)$$

where $h_s(t) \geq 0$ is a time-varying (differentiable) function and $(1 - M_s(t))$ ensures the intensity of endogenous events becomes zero if the story is verified (*i.e.*, fact checked).

For endogenous events, we consider the following form for the intensity functions:

$$\lambda_s^e(t) = \Big[\overbrace{\int_0^t r_s(\tau)g(t - \tau)dN_s^e(\tau)}^{\text{Reshares}}$$
$$+ \overbrace{\int_0^t g(t - \tau)dN_s^p(\tau)}^{\text{Posts}} \Big] \overbrace{(1 - M_s(t))}^{\text{Fact check}}, \quad (2)$$

where the first term, with $r_s(\tau) \sim \text{Bernoulli}(r_s)$, models the exposures due to reshares of story s by the previous users, the second term models the exposures due to previous posts of story s by users on their own initiative (*i.e.*, exogenous events). In the first two terms, $g_s(t)$ denotes an exponential triggering kernel, $g(t) := \gamma \exp(-\omega t)\mathbb{I}(t \geq 0)$, which models the decay of *influence* of previous reshares and posts over time. By making the intensity dependent on the history of reshares and posts, it becomes a stochastic process by itself.

Given the above functional form for the intensity of endogenous events, the following alternative representation based on stochastic differential equations (SDEs) with jumps, which can be derived using Ito's calculus [16], will be useful to design our algorithm:

PROPOSITION 2.1. *Let $N_s^e(t)$ be a counting process with associated intensity $\lambda_s^e(t)$, given by Eq. 2. The tuple $(N_s^e(t), \lambda_s^e(t))$ is a doubly stochastic Markov process whose dynamics can be represented by the following SDE with jumps:*

$$d\lambda_s^e(t) = -\omega\lambda_s^e(t)dt - \lambda_s^e(t)dM_s(t)$$
$$+ \gamma \Big[r_s(t)dN_s^e(t) + (1 - M(t))dN_s^p(t) \Big] \quad (3)$$

Estimated rate of misinformation. If one cannot send all stories for fact checking, ideally, one may like to send only fake stories and favor those which, if not fact checked, would reach the greatest number of users. However, we cannot directly observe whether a story is fake (that is why we send it for fact checking!) and we do not know the total number of users who will be exposed to the story if not fact checked. Instead, for each story, we will leverage the users' flags and exposures to compute a running estimate of the rate of misinformation due to that story. Then, we will find the optimal fact checking intensities $\mathbf{u}(t)$ that minimize a nondecreasing function of the estimated misinformation rates over time.

Given a story $s \in \mathcal{S}$, the number of users $N_s^f(t)$ who flagged the story up to time t can be formally defined as

$$N_s^f(t) := \int_0^t f_s(\tau)dN_s^e(\tau),$$

where $N_s^e(t)$ counts the number of users exposed to story s by time t, $f_s(\tau) \sim f_s$, and it readily follows that $dN_s^f(t) = f_s(t)dN_s^e(t)$. Then, we can estimate the average number of users $\bar{N}_s^m(t)$ exposed to misinformation by time t, conditioned on the number of exposed users and flags, as

$$\bar{N}_s^m(t) := p_{m|s,f=1}N_s^f(t) + p_{m|s,f=0}(N_s^e(t) - N_s^f(t))$$
$$= p_{m|f=1}N_s^f(t) + p_{m|f=0}(N_s^e(t) - N_s^f(t)),$$

where we assume that the probability that a story is misinformation given that a user did or did not flag it is equal for all stories, *i.e.*, $p_{m|s,f} = p_{m|f}$, and one can estimate $p_{m|f}$ from historical flag and exposure data about fact checked stories. Here, the probability $p_{m|f}$ characterizes how *good* the crowd is at spotting misinformation[10].

Next, we can compute the differential $d\bar{N}_s^m(t)$ as

$$d\bar{N}_s^m(t) = \left[(p_{m|f=1} - p_{m|f=0})f_s(t) + p_{m|f=0}\right]dN_s^e(t)$$

and define the rate of misinformation as

$$\lambda_s^m(t)dt = \mathbb{E}[d\bar{N}_s^m(t)]$$
$$= \left[(p_{m|f=1} - p_{m|f=0})f_s + p_{m|f=0}\right]\lambda_s^e(t)dt. \qquad (4)$$

Unfortunately, the flagging probability f_s may depend on many complex factors, *e.g.*, content, source, and is generally unknown. To overcome this challenge, we assume a Beta prior on f_s, *i.e.*, $f_s \sim Beta(\alpha, \beta)$, and compute instead a posterior estimate of the rate of misinformation, which leverages both the number of exposures $N_s^e(t)$ and flags $N_s^f(t)$ by time t, *i.e.*,

$$\hat{\lambda}_s^m(t)dt = \mathbb{E}_{f_s(t),f_s}[d\bar{N}_s^m(t)]$$
$$= \left[(p_{m|f=1} - p_{m|f=0})\frac{\alpha + N_s^f(t)}{\alpha + \beta + N_s^e(t)}\right.$$
$$\left. + p_{m|f=0}\right]\lambda_s^e(t)dt, \qquad (5)$$

where we used the conjugacy between the Bernoulli and the Beta distributions, *i.e.*,

$$f_s|N_s^f, N_s^e \sim Beta\left(\alpha + N_s^f(t), \beta + N_s^e(t) - N_s^f(t)\right).$$

Finally, given a set of stories $\mathcal{S}$, we denote the vector of posterior estimates of their associated rate of misinformation as $\hat{\boldsymbol{\lambda}}^m(t) = [\hat{\lambda}_s^m(t)]_{s\in\mathcal{S}}$.

The fact checking scheduling problem. Given a set of unverified stories $\mathcal{S}$, we aim to find the optimal fact checking intensities $\boldsymbol{u}(t)$ that minimize the expected value of nondecreasing convex loss function $\ell(\hat{\boldsymbol{\lambda}}^m(t), \boldsymbol{u}(t))$ of the posterior estimates of the stories rates of misinformation $\hat{\boldsymbol{\lambda}}^m(t)$ and the fact checking intensities $\boldsymbol{u}(t)$ over a time window $(t_0, t_f]$, *i.e.*,

$$\underset{\boldsymbol{u}(t_0, t_f]}{\text{minimize}} \quad \mathbb{E}\left[\phi(\hat{\boldsymbol{\lambda}}^m(t_f)) + \int_{t_0}^{t_f} \ell(\hat{\boldsymbol{\lambda}}^m(\tau), \boldsymbol{u}(\tau))d\tau\right]$$
$$\text{subject to} \quad \boldsymbol{u}(t) \geq 0 \quad \forall t \in (t_0, t_f], \qquad (6)$$

---

[10]One could consider different false positive and false negative rates per user and per story, *e.g.*, using text analysis or domain knowledge about the website that published the story. However, for simplicity, we leave such extensions for future work.

where $\boldsymbol{u}(t_0, t_f]$ denotes the fact checking intensities from t_0 to t_f, the expectation is taken over all possible realizations of the marked temporal point processes associated to the endogenous and exogenous events of all stories from t_0 to t_f, and $\phi(\hat{\boldsymbol{\lambda}}^m(t_f))$ is an arbitrary penalty function. Here, by considering a nondecreasing loss on both the posterior estimate of the rates of misinformation and fact checking intensities, we penalize high levels of misinformation and we limit the number of stories that are sent for fact checking.

3 PROPOSED ALGORITHM

In this section, we find the optimal fact-checking intensities $\boldsymbol{u}(t)$ that minimizes Eq. 6 from the perspective of stochastic optimal control of jump SDEs [16]. To ease the exposition, we first derive a solution for one story, introduce an efficient algorithm that implements the solution, and then generalize the solution and the efficient algorithm to multiple stories.

Optimizing for one story. Given an unverified story s with fact checking reviewing intensity $u_s(t) = u(t)$, exogenous intensity $\lambda_s^p(t) = \lambda^p(t)$, endogenous intensity $\lambda_s^e(t) = \lambda^e(t)$ and associated counting processes $M_s(t) = M(t)$, $N_s^p(t) = N^p(t)$ and $N_s^e(t) = N^e(t)$, respectively, we can rewrite the fact checking scheduling problem defined in Eq. 6 as:

$$\underset{u(t_0, t_f]}{\text{minimize}} \quad \mathbb{E}\left[\phi(\hat{\lambda}^m(t_f)) + \int_{t_0}^{t_f} \ell(\hat{\lambda}^m(\tau), u(\tau))d\tau\right]$$
$$\text{subject to} \quad u(t) \geq 0 \quad \forall t \in (t_0, t_f], \qquad (7)$$

where the dynamics of $N^e(t)$ are given by Eq. 3 and the posterior estimate of the misinformation rate is given by Eq. 5. Note that we are now focusing on a single story rather than on multiple stories.

Next, we define an optimal cost-to-go function J for the above problem, use Bellman's principle of optimality to derive the associated Hamilton-Jacobi-Bellman (HJB) equation [3], and exploit the structure of our problem to find a solution to the HJB equation.

Definition 3.1. The optimal cost-to-go J is defined as the minimum of the expected value of the cost of going from state $(M(t), N^e(t), N^f(t), N^p(t), \lambda^e(t))$ at time t to the final state at time t_f, *i.e.*,

$$J(M(t), N^e(t), N^f(t), N^p(t), \lambda^e(t), t) \qquad (8)$$
$$= \min_{u(t, t_f]} \mathbb{E}\left[\phi(\hat{\lambda}^m(t_f)) + \int_t^{t_f} \ell(\hat{\lambda}^m(\tau), u(\tau))d\tau\right].$$

Now, using the Markov property of the state variables $(M(t), N^e(t), N^f(t), N^p(t), \lambda^e(t))$, we can apply Bellman's principle of optimality to the above definition to break the problem into smaller subproblems and rewrite Eq. 8 as

$$0 = \min_{u(t, t+dt]} \left\{ \mathbb{E}\left[dJ(M(t), N(t), N^f(t), N^p(t), \lambda^e(t), t)\right] \right.$$
$$\left. + \ell(\hat{\lambda}^m(t), u(t))dt \right\}. \qquad (9)$$

Next, we differentiate J with respect to time t, $M(t)$, $N^e(t)$, $N^f(t)$, $N^p(t)$ and $\lambda^e(t)$ using Lemma 5.1 (in the Appendix) with $x(t) =$

$\lambda^e(t), y(t) = \lambda^p(t) = h(t), z(t) = r(t), w(t) = f(t))$ and $F = J$.

$$
\begin{aligned}
dJ = &\big[J(M, N^e + 1, N^f + 1, N^p, \lambda^e + \gamma, t) f(t) r(t) \\
&+ J(M, N^e + 1, N^f + 1, N^p, \lambda^e, t) f(t)(1 - r(t)) \\
&+ J(M, N^e + 1, N^f, N^p, \lambda^e + \gamma, t)(1 - f(t)) r(t) \\
&+ J(M, N^e + 1, N^f, N^p, \lambda^e, t)(1 - f(t))(1 - r(t)) \\
&- J(M, N^e, N^f, N^p, \lambda^e, t) \big] dN^e(t) \\
&+ \big[J(M, N^e, N^f, N^p + 1, \lambda^e + \gamma, t) \\
&- J(M, N^e, N^f, N^p, \lambda^e, t) \big] dN^p(t) \\
&- \big[J(M, N^e, N^f, N^p, \lambda^e, t) - J(M + 1, N^e, N^f, N^p, 0, t) \big] dM(t) \\
&+ J_t - \omega \lambda^e(t) J_{\lambda^e},
\end{aligned}
$$

Then, using that $\mathbb{E}[dN^e(t)] = \lambda^e(t)dt, \mathbb{E}[dN^p(t)] = h(t)dt, \mathbb{E}[dM(t)]$ $= (1 - M(t))u(t)dt, \mathbb{E}[r(t)] = r_s = r$, and $\mathbb{E}[f(t)] = \frac{\alpha + N^f(t)}{\alpha + \beta + N^e(t)}$, the HJB equation follows:

$$
\begin{aligned}
0 = &\big[J(M, N^e + 1, N^f + 1, N^p, \lambda^e + \gamma, t) \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)} r \\
&+ J(M, N^e + 1, N^f + 1, N^p, \lambda^e, t) \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)} (1 - r) \\
&+ J(M, N^e + 1, N^f, N^p, \lambda^e + \gamma, t)(1 - \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)}) r \\
&+ J(M, N^e + 1, N^f, N^p, \lambda^e, t)(1 - \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)})(1 - r) \\
&- J(M, N^e, N^f, N^p, \lambda^e, t) \big] \lambda^e(t) \\
&+ \big[J(M, N^e, N^f, N^p + 1, \lambda^e + \gamma, t) \\
&- J(M, N^e, N^f, N^p, \lambda^e, t) \big] (1 - M(t)) h(t) \\
&+ \min_{u(t, t_f]} \Big\{ l(\hat{\lambda}^m(t), u(t)) + J_t - \omega \lambda^e(t) J_{\lambda^e} \\
&\qquad - \big[J(M, N^e, N^f, N^p, \lambda^e, t) \\
&\qquad - J(M + 1, N^e, N^f, N^p, 0, t) \big] (1 - M(t)) u(t) \Big\}, \quad (10)
\end{aligned}
$$

with $J(M(t_f), N^e(t_f), N^f(t_f), N^p(t_f), \lambda^e(t_f), t_f) = \phi(\hat{\lambda}^e(t_f))$ as the terminal condition.

To solve the above HJB equation, we need to define the penalty function ϕ and the loss function ℓ. Following the literature on stochastic optimal control [16], we consider the following quadratic forms, which penalize high levels of misinformation and limit the number of stories that are sent for fact checking:

$$
\phi(\hat{\lambda}^m(t_f)) = \frac{1}{2}(\hat{\lambda}^m(t_f))^2
$$

$$
\ell(\hat{\lambda}^m(t), u(t)) = \frac{1}{2}(\hat{\lambda}^m(t))^2 + \frac{1}{2}qu^2(t),
$$

where q is a tunable parameter to trade-off between the number of stories sent for fact checking and the spread of misinformation. With the loss function, we take the derivative with respect to $u(t)$ in Eq. 10 and uncover the relationship between the optimal fact checking intensity and the optimal cost J:

$$
\begin{aligned}
u^*(t) = q^{-1}(1 - M(t)) \big[&J(M(t), N^e(t), N^f(t), N^p(t), \lambda^e(t), t), \\
&- J(M(t) + 1, N^e(t), N^f(t), N^p(t), 0, t) \big]. \quad (11)
\end{aligned}
$$

Then, we plug in the above expression in Eq. 10 and find a solution to the resulting nonlinear differential equation using the following Lemma:

LEMMA 3.2. *The optimal cost-to-go J that satisfies the HJB equation, defined by Eq. 10, is given by:*

$$
\begin{aligned}
&J(M(t), N^e(t), N^f(t), N^p(t), \lambda^e(t), t) \\
&= q^{\frac{1}{2}} \big[(p_{m|f=1} - p_{m|f=0}) \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)} + p_{m|f=0} \big] \big[\lambda^e(t) \\
&\qquad - \gamma N^p(t) - (\gamma r - \omega)(\alpha + \beta + N^e(t)) \big].
\end{aligned}
$$

Proof Sketch. We verify the above cost-to-go J satisfies the HJB equation, given by Eq. 10. To do so, we use that $J_t = 0$ and

$$
J_{\lambda^e} = q^{\frac{1}{2}} \big[(p_{m|f=1} - p_{m|f=0}) \tfrac{\alpha + N^f(t)}{\alpha + \beta + N(t)} + p_{m|f=0} \big].
$$

Here, note that the HJB equation needs to be satisfied only for $M(t) \in \{0, 1\}$, which are the only feasible values for the counting process associated to the fact checking event. Next, we use the above Lemma to recover the optimal fact checking intensity $u^*(t)$:

THEOREM 3.3. *Given a story s, the optimal fact checking intensity for the fact checking scheduling problem, defined by Eq. 7, under quadratic loss is given by:*

$$
u^*(t) = q^{-\frac{1}{2}}(1 - M(t)) \Big[p_{m|f=0} \\
+ (p_{m|f=1} - p_{m|f=0}) \left(\frac{\alpha + N^f(t)}{\alpha + \beta + N^e(t)} \right) \Big] \lambda^e(t). \quad (12)
$$

The above result reveals a linear relationship between the optimal fact checking intensity and the endogenous (*i.e.*, exposure) intensity $\lambda^e(t)$. Moreover, the corresponding coefficient depends on the number of user exposures $N^e(t)$ and the number of flags $N^f(t)$, and can increase and decrease over time. Remarkably, the optimal intensity does not depend on the exogenous intensity $\lambda^p(t)$, which is generally unknown.

Finally, we derive an efficient procedure to sample from the above intensity, summarized in Algorithm 1, which we name CURB. At a high level, as a story spreads through out the network, the algorithm updates its belief about the best time to fact-check based on how quickly the story is spreading as well as what is the likelihood that the story is misinformation. Within the algorithm, $Next()$ returns the next endogenous (*i.e.*, exposure) event, which occurs at time t' with resharing and flagging indicators r and f, respectively, $Update(N^e(t), N^f(t), \lambda^e(t))$ updates $u(t)$ using Eq. 12, and $Sample(\tau, u(t))$ samples from an intensity $u(t)$ starting at time τ using a standard thinning algorithm [21]. If the fact checking intensity increases, by $r = 1$ or $f = 1$, we apply the superposition theorem [18], *i.e.*, we sample a new candidate fact checking time κ and we take the minimum between κ and the previous candidate fact checking time. If the fact checking intensity decreases, by $f = 0$, we first use a thinning argument to decide whether to keep the current candidate fact checking time τ or we sample a new candidate fact checking time, greater than τ. This sampling procedure is very efficient since it only needs to sample $O(N^e(t_f))$ times from an inhomogeneous Poisson process. Finally, note that if we gather more and more evidence that a story s is not misinformation ($f = 0$ for many exposures), the candidate fact checking time τ will eventually become greater than t_f and the story will never be sent for fact checking.
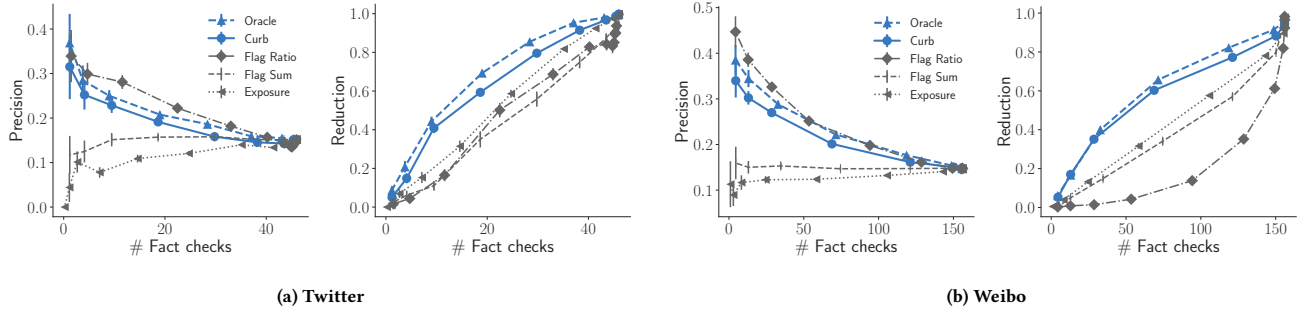
(a) Twitter

(b) Weibo

Figure 1: Performance vs. number of fact checking events. We measure performance in terms of misinformation reduction, which is the fraction of unverified exposures that fact checking prevented, and precision, which is the fraction of fact checked stories that are fake. Oracle and CURB achieve a comparable performance in both datasets, beating the threshold baseline by large margins.

Algorithm 1: The CURB Algorithm

Input: Parameters q, α, β, $p_{m|f=1}$, $p_{m|f=0}$, t_f

Initialization : $N(t) \leftarrow 0$; $N^f(t) \leftarrow 0$; $\lambda^e(t) \leftarrow 0$;
$\qquad\qquad Update(N^e(t), N^f(t), \lambda^e(t))$

Output: Fact checking time τ

$\tau \leftarrow t_f$

$(t', r, f) \leftarrow Next()$

while $t' < \tau$ **do**

$\quad u_0(t) \leftarrow u(t)$

$\quad N^e(t) \leftarrow N^e(t) + 1; \quad N^f(t) \leftarrow N^f(t) + f$

$\quad u(t) \leftarrow Update(N^e(t), N^f(t), \lambda^e(t))$

$\quad$**if** $f = 0$ **then**

$\quad\quad x \leftarrow Uniform(0, 1)$

$\quad\quad$**if** $u(t)/u_0(t) < x$ **then**

$\quad\quad\quad \tau \leftarrow Sample(\tau, u(t))$

$\quad\quad$**end**

$\quad$**end**

$\quad$**if** $r = 1$ **then**

$\quad\quad \lambda^e(t) \leftarrow \lambda^e(t) + g(t - t')$

$\quad\quad u(t) \leftarrow Update(N^e(t), N^f(t), \lambda^e(t))$

$\quad$**end**

$\quad \kappa = Sample(t', \max(0, u(t) - u_0(t)))$

$\quad \tau \leftarrow min(\tau, \kappa)$

$\quad (t', r, f) \leftarrow Next()$

end

return τ

Optimizing for multiple stories. Given a set of unverified stories $\mathcal{S}$ with fact checking intensities $\boldsymbol{u}(t)$, exogenous intensities $\boldsymbol{\lambda}^p(t)$, endogenous intensities $\boldsymbol{\lambda}^e(t)$ and associated counting processes $\boldsymbol{M}(t)$, $\boldsymbol{N}^p(t)$ and $\boldsymbol{N}^e(t)$, we can solve the fact checking scheduling problem defined by Eq. 6 similarly as in the case of a single story. In particular, consider the following quadratic form for the penalty function ϕ and loss function ℓ:

$$\ell(\hat{\boldsymbol{\lambda}}^m(t), \boldsymbol{u}(t)) = \frac{1}{2} \sum_{s \in \mathcal{S}} (\hat{\lambda}_s^m(t))^2 + \frac{1}{2} \sum_{s \in \mathcal{S}} q_s u_s^2(t),$$

where $\{q_s\}_{s \in \mathcal{S}}$ are given parameters, which trade off the number of stories sent for fact checking and the spread of misinformation,

and may favor fact checking some stories over others. Then, we can derive the optimal control signal intensity for each story, using the independence assumption between stories, proceeding similarly as in the case of a single story.

THEOREM 3.4. *Given a set of stories $\mathcal{S}$, the optimal fact checking intensity for each story $s \in \mathcal{S}$, which minimizes Eq. 6, under quadratic loss and penalty functions is given by:*

$$u_s^*(t) = q_s^{-\frac{1}{2}} (1 - M_s(t)) \Big[p_{m|f=0}$$
$$+ (p_{m|f=1} - p_{m|f=0}) \left(\frac{\alpha + N_s^f(t)}{\alpha + \beta + N_s^e(t)} \right) \Big] \lambda_s^e(t). \quad (13)$$

Finally, we can sample the fact checking times for all stories by running $|\mathcal{S}|$ instances of CURB, one per story, and the overall number of required samples is $O(\sum_{s \in \mathcal{S}} N_s^e(t_f))$. Moreover, note that the instances can be run in parallel and thus our algorithm is highly scalable.

4 EXPERIMENTS

In this section, we evaluate our algorithm on data gathered from two social networking sites, Twitter and Weibo, and compare its performance with three baselines via two metrics.

Dataset description and experimental setup. We use data gathered from Twitter and Weibo as reported in previous work [20, 28], which comprises posts and reshares for a variety of (manually annotated) genuine and fake stories[11], respectively.

More specifically, the Twitter dataset contains 192,350 posts and reshares from 117,824 users for 111 unique stories. The Weibo dataset contains 3,752,459 posts and reshares from 2,819,338 users for 4,663 unique stories. We filtered out stories posted or reshared more than 3,000 times as well as stories whose number of posts or reshares taking place after the last decile of the observation period is greater than 1%. Finally, we filtered out fake stories at random until the percentage of fake stories is less than 15%[12].

---

[11]In the Twitter and Weibo datasets, stories were fact checked using snopes.com and the Sina community management center, service.account.weibo.com.

[12]We subsample the number of fake stories to resemble a more realistic scenario in which the number of fake stories is *small* in comparison with the total number of stories. We obtained qualitatively similar results without subsampling.
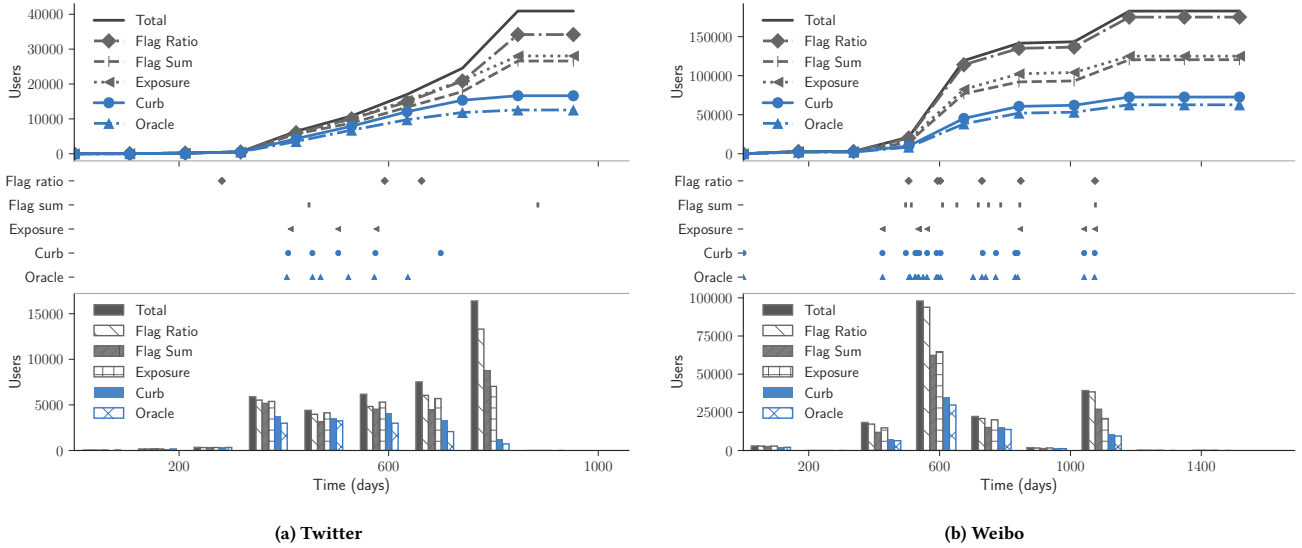
(a) Twitter

(b) Weibo

Figure 2: Misinformation reduction vs. time. For both (a) and (b), the bottom panel shows the number of users exposed to misinformation at each time period, the middle panel shows when different methods send stories for fact checking, and the top panel shows the cumulative number of users. The bars and lines show the different fact checking schedules and "Total" which shows the numbers in absence of fact checking. Both Curb and Oracle are able to prevent the spread of misinformation before it becomes viral, whereas the other baselines cannot do so. The parameters for all methods are set so that the total number of fact-checked stories is 15 ± 4 for Twitter and 64 ± 10 for Weibo.

After these preprocessing steps, our Twitter dataset consists of 28,486 posts and reshares from 18,880 users for 7 fake stories and 39 genuine stories and our Weibo dataset consists of 93,943 posts and reshares from 88,913 users for 23 fake stories and 133 genuine stories. Unfortunately, the datasets do not contain any information about the timing (or number) of exposures nor flags. To remedy the data unavailability, we generate user exposures based on user reshares and sample flags from Bernoulli distributions. We experiment with different parameter settings to ensure that our model's efficacy is robust throughout different settings. The details of the data generation steps are described in the following paragraph.

For each story, we sample exposure events using the exposure intensity defined by Eq. 2, which does depend on the observed posts and reshares. There, we set $\gamma = 10^{-4}$ and $\omega = 10^{-5}$. This choice of parameters results in approximately 10 to 20 exposures per post (or reshare) and a half-life of approximately 19 hours per post (or reshare). Under this setting, our Twitter and Weibo datasets contain 313,357 and 1,054,449 exposures respectively. Unless otherwise stated, for each exposure, we sample flags for fake and genuine stories using two bernoulli distributions with $p_{f=1|m=1} = 0.3$ and $p_{f=1|m=0} = 0.01$, respectively. Moreover, we estimate p_d using the ratio between fake and overall number of stories, compute $p_{m|f=1}$ and $p_{m|f=0}$ from $p_{f=1|m}, p_{f=0|m}$, and p_d, and set $\alpha/(\alpha + \beta) = (p_{f=1|d} + p_{f=0|d})p_d$. Finally, note that we obtained qualitatively similar results for other experimental settings.

Evaluation metrics and baselines. Throughout this section, we evaluate the performance via two metrics: precision and misinformation reduction. Precision is the fraction of fact checked stories that are fake. Misinformation reduction is the fraction of unverified exposures that fact checking prevented.

We compare the performance of Curb against an "Oracle", which is a variant of Curb and has access to the true flag probability p_f, and three baselines. The first baseline ("Flag Ratio") samples the fact checking time for each story using an intensity which is proportional to the ratio between the number of flags and the total number of exposures, *i.e.*, $u_s(t) = q_s(\alpha + N_s^f(t))/(\alpha + \beta + N_s^e(t))$. The second baseline ("Flag Sum") sends a story for fact checking as soon as it accumulates a certain number of flags. The third baseline ("Exposure") samples the fact checking time for each story using an intensity which is proportional to the exposure intensity, *i.e.*, $u_s(t) = q_s \lambda_s^e(t)$. Here, note that the Flag Sum baseline utilizes a deterministic policy while our method, the Oracle and the other baselines use a stochastic policy defined by an intensity function.

Solution quality. We first evaluate the performance of our method, the oracle and the three baselines against the number of fact checking events. Figure 1 summarizes the results, which show that: (i) the more the number of fact checking events, the higher the average misinformation reduction for all methods; and, (ii) Oracle and Curb achieve a comparable performance in both datasets, outperforming the baselines in most cases. Next, we evaluate the performance in terms of misinformation reduction over time in Figure 2. We find that both Curb and the Oracle are able to prevent the spread of
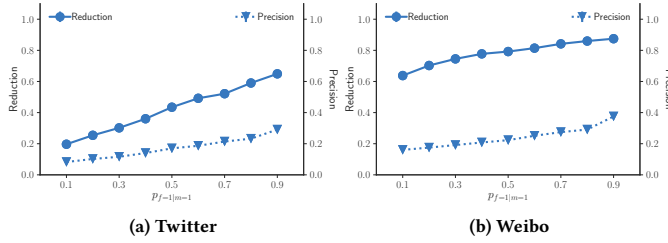
(a) Twitter

(b) Weibo

Figure 3: Performance for different true positive rates, $p_{f=1|m=1}$. Here, we set the false positive rate to $p_{f=1|m=0} = 0.01$. The higher the true positive rate, the *better* the crowd is at spotting misinformation and the more effective CURB is. Total number of fact checked stories is 21 ± 2 for Twitter and 90 ± 15 for Weibo.

misinformation before it becomes viral. In contrast, the baselines are unable to spot misinformation that will become viral in time partly due to ignoring either the flags or the exposure dynamics.

Flagging behavior and misinformation evidence. In this section, we first investigate how sensitive our algorithm is to the crowd's flagging behavior and then explore the influence of the hyperparameters α and β on the posterior estimate of the misinformation rate.

Intuitively, the more (less) accurate the crowd is at spotting misinformation, the more (less) effective our algorithm will be. Figure 3 confirms this intuition by showing the performance of our algorithm for different true positive rates, $p_{f=1|m=1}$.

Next, we investigate the influence of the hyperparameter α on the posterior estimate of the misinformation rate $\lambda_s^m(t)$, which the optimal intensity $u^*(t)$ depends on, and the posterior estimate of the flagging probability $\mathbb{E}[f_s|N_s^e(t), N_s^f(t)] = \hat{f}_s$ by examining an example of a genuine and a fake story. Figure 4 summarizes the results, which show that: (i) the higher the value of α, the smaller the effect of individual exposures and flags on the posterior estimate of both the misinformation rate and the flagging probability when a story starts spreading—the lower the variance of the estimates over time; and, (ii) the posterior estimate of the misinformation rate, which the optimal intensity $u^*(t)$ depend on, changes in a more intricate way than the posterior estimate of the flagging probability since it combines the latter with the exposure intensity. This enables our algorithm to be more cautious with viral stories even if the evidence of misinformation is weak.

5 CONCLUSIONS

In this paper, we have introduced an efficient online algorithm, CURB, that leverages the crowd to detect and prevent the spread of fake news and misinformation in online social networking sites. In doing so, we establish an unexplored connection between stochastic online optimal control of SDEs with jumps, survival analysis, and Bayesian inference. We experimented with two real-world datasets gathered from Twitter and Weibo and showed that our algorithm effectively reduces the spread of misinformation, outperforms several competitive baselines and is robust to different user flagging behaviors.

There are many interesting directions for future work. For example, we assumed every person in the crowd is equally good (or bad) at flagging misinformation. It would be interesting to relax this assumption, infer each person's trustworthiness, and design algorithms that are robust to adversarial behavior from part of the crowd. Also, it would be worthwhile to investigate if one's network position affects the susceptibility of being exposed to fake news. For example, a subgroup of biased people may be the main source of misinformation. If that is the case, explicitly incorporating the network structure in the model would be beneficial in mitigating the spread of misinformation. Moreover, we considered that stories are independent and the probability that a story is misinformation given that a user did or did not flag it is equal for all stories. However, stories may be dependent and the probability that a story is misinformation given a user did or did not flag it may be different for stories supported by different sources (or domains) [36]. Finally, our algorithm optimizes a quadratic loss of the posterior estimate of the misinformation rate, however, it would be useful to derive fact checking intensities for other losses capturing more nuanced goals.

APPENDIX

LEMMA 5.1. *Let $x(t)$ be a jump-diffusion process defined by the following SDE:*

$$dx(t) = f(x(t), t)dt + g(x(t), t)z(t)dN^e(t)$$
$$+ h(x(t), t)(1 - M(t))dN^p(t) + k(x(t), t)dM(t),$$

where $M(t)$, $N^e(t)$, $N^p(t)$ are independent jump processes, $N_f(t)$ is a jump process whose increment is defined as $dN_f(t) = w(t)dN^e(t)$, $w(t) \in \{0, 1\}$, and $z(t) \in \{0, 1\}$. If the function $F(M(t), N^e(t), N^f(t), N^p(t), x(t), t)$ is once continuously differentiable in $x(t)$ and t, then,

$$dF(M(t), N(t), N^f(t), N^p(t), x(t), t)$$
$$= \Big[F(M, N^e + 1, N^f + 1, N^p, x + g, t)w(t)z(t)$$
$$+ F(M, N^e + 1, N^f + 1, N^p, x, t)w(t)(1 - z(t))$$
$$+ F(M, N^e + 1, N^f, N^p, x + g, t)(1 - w(t))(z(t))$$
$$+ F(M, N^e + 1, N^f, N^p, x, t)(1 - w(t))(1 - z(t))$$
$$- F(M, N^e, N^f, N^p, x, t) \Big] dN^e(t)$$
$$+ \Big[F(M, N^e, N^f, N^p + 1, x + h, t)$$
$$- F(M, N^e, N^f, N^p, x, t) \Big] dN^p(t)$$
$$+ \Big[F(M + 1, N^e, N^f, N^p, x + k, t)$$
$$- F(M, N^e, N^f, N^p, x, t) \Big] dM(t) + F_t dt + f F_x dt,$$

where for notational simplicity we dropped the arguments of the functions f, g, h, k.

Proof Sketch. The differential of J can be found using Ito's calculus [16]. In particular, using that the bilinear forms $dtdN^e(t) = dtdN^p(t) = dtdM(t) = 0$ and $dN^e(t)dM(t) = 0$ and $N^p(t)dM(t) = 0$ by the zero-one jump law.
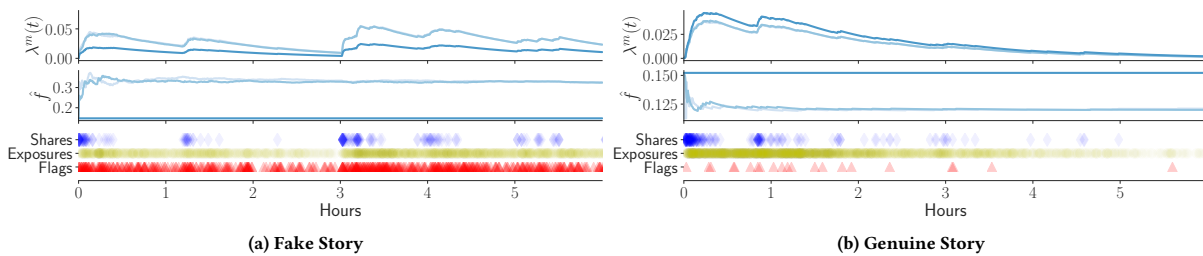
(a) Fake Story

(b) Genuine Story

Figure 4: Influence of the hyperparameter α on the posterior estimate of the misinformation rate, which the optimal intensity $u_s^*(t)$ depends on, and the posterior estimate of the flagging probability. In the top and middle panels, darker shades indicate stronger prior. The higher the value of α, the smaller the effect of individual exposures and flags on the posterior estimate of both the misinformation rate and the flagging probability.

REFERENCES

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

[2] B. T. Adler and L. De Alfaro. 2007. A content-driven reputation system for the Wikipedia. In *WWW*.

[3] D. P. Bertsekas. 1995. *Dynamic programming and optimal control*. Athena Scientific.

[4] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. 2005. Link analysis ranking: algorithms, theory, and experiments. *TOIT* 5, 1 (2005), 231–297.

[5] L. Chen, Z. Yan, W. Zhang, and R. Kantola. 2015. TruSMS: a trustworthy SMS spam control system based on trust management. *Future Generation Computer Systems* 49 (2015), 77–93.

[6] P. Chia and S. Knapskog. 2011. Re-evaluating the wisdom of crowds in assessing web security. In *International Conference on Financial Cryptography and Data Security*.

[7] G. L. Ciampaglia, P. S., L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PLOS ONE* 10, 6 (06 2015), 1–13.

[8] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez. 2016. Learning and Forecasting Opinion Dynamics in Social Networks. In *NIPS*.

[9] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. 2014. From data fusion to knowledge fusion. In *VLDB*.

[10] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. 2015. Knowledge-based trust: Estimating the trustworthiness of web sources. In *VLDB*.

[11] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. 2014. Shaping social activity by incentivizing users. In *NIPS*.

[12] D. Freeman. 2017. Can You Spot the Fakes?: On the Limitations of User Feedback in Online Social Networks. In *WWW*.

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

[14] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *SocInfo*.

[15] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. 2004. Combating web spam with trustrank. In *VLDB*.

[16] F. B. Hanson. 2007. *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*. Society for Industrial and Applied Mathematics.

[17] N. Hung, D. Thang, M. Weidlich, and K. Aberer. 2015. Minimizing Efforts in Validating Crowd Answers. In *SIGMOD*.

[18] J. F. C. Kingman. 1993. *Poisson processes*. Wiley Online Library.

[19] S. Kumar, R. West, and J. Leskovec. 2016. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *WWW*.

[20] S. Kwon, M. Cha, and K. Jung. 2017. Rumor detection over varying time windows. *PLOS ONE* 12, 1 (2017), e0168344.

[21] P. A. Lewis and G. S. Shedler. 1979. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics* 26, 3 (1979), 403–413.

[22] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han. 2015. On the discovery of evolving truth. In *KDD*.

[23] M. Liu, L. Jiang, J. Liu, X. Wang, J. Zhu, and S. Liu. 2017. Improving Learning-from-Crowds through Expert Validation. In *IJCAI*.

[24] X. Liu, X. L. Dong, B. C. Ooi, and D. Srivastava. 2011. Online data fusion. In *VLDB*.

[25] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah. 2015. Real-time rumor debunking on twitter. In *CIKM*.

[26] M. Lukasik, P. K. Srijith, D. Vu, K. Bontcheva, A. Zubiaga, and T. Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *ACL*.

[27] C. Lumezanu, N. Feamster, and H. Klein. 2012. # bias: Measuring the tweeting behavior of propagandists. In *ICWSM*.

[28] J. Ma, W. Gao, P. Mitra, S. Kwon, B. Jansen, K. Wong, and M. Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *IJCAI*.

[29] M. Mendoza, B. Poblete, and C. Castillo. 2010. Twitter Under Crisis: Can we trust what we RT?. In *Workshop on Social Media Analytics*.

[30] T. Moore and R. Clayton. 2008. Evaluating the wisdom of crowds in assessing phishing websites. *Lecture Notes in Computer Science* 5143 (2008), 16–30.

[31] A. Pal, V. Rastogi, A. Machanavajjhala, and P. Bohannon. 2012. Information integration over time in unreliable and uncertain environments. In *WWW*.

[32] J. Pasternack and D. Roth. 2013. Latent credibility analysis. In *WWW*.

[33] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *EMNLP*.

[34] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. 2011. Truthy: mapping the spread of astroturf in microblog streams. In *WWW*.

[35] N. Ruchansky, S. Seo, and Y. Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *CIKM*.

[36] B. Tabibian, I. Valera, M. Farajtabar, L. Song, B. Schoelkopf, and M. Gomez-Rodriguez. 2017. Distilling Information Reliability and Source Trustworthiness from Digital Traces. In *WWW*.

[37] M. Tanushree, G. Wright, and E. Gilbert. 2017. Parsimonious language model of social media credibility across disparate events. In *CSCW*.

[38] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *ACL*.

[39] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Zhao. 2012. Social Turing Tests: Crowdsourcing Sybil Detection. In *arXiv*.

[40] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher. 2014. Towards cyber-physical systems in social spaces: The data reliability challenge. In *RTSS*.

[41] Y. Wang, W. Grady, E. Theodorou, and L. Song. 2017. Variational Policy for Guiding Point Processes. In *ICML*.

[42] Y. Wang, G. Williams, E. Theodorou, and L. Song. 2017. Variational Policy for Guiding Point Processes. In *ICML*.

[43] W. Wei and X. Wan. 2017. Learning to Identify Ambiguous and Misleading News Headlines. In *IJCAI*.

[44] M. Wu and A. Marian. 2007. Corroborating Answers from Multiple Web Sources.. In *WebDB*.

[45] H. Xiao, J. Gao, Q. Li, F. Ma, L. Su, Y Feng., and A. Zhang. 2016. Towards Confidence in the Truth: A Bootstrapping based Truth Discovery Approach. In *KDD*.

[46] X. Yin and W. Tan. 2011. Semi-supervised truth discovery. In *WWW*.

[47] A. Zarezade, A. De, H. Rabiee, and M. Gomez-Rodriguez. 2017. Cheshire: An Online Algorithm for Activity Maximization in Social Networks. In *Allerton Conference*.

[48] A. Zarezade, U. Upadhyay, H. Rabiee, and M. Gomez-Rodriguez. 2017. RedQueen: An Online Algorithm for Smart Broadcasting in Social Networks. In *WSDM*.

[49] B. Zhao and J. Han. 2012. A probabilistic model for estimating real-valued truth from conflicting sources. In *QDB*.

[50] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. 2012. A bayesian approach to discovering truth from conflicting sources for data integration. In *VLDB*.

[51] Z. Zhao, P. Resnick, and Q. Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW*.

[52] E. Zheleva, A. Kolcz, and L. Getoor. 2008. Trusting spam reporters: A reporter-based reputation system for email filtering. *TOIS* 27, 1 (2008), 3.