

Steering Opinion Dynamics in Information Diffusion Networks

Yichen Wang[◊], Evangelos Theodorou*, Apurv Verma[◊], and Le Song[◊]

[◊]College of Computing, Georgia Institute of Technology

*School of Aerospace Engineering, Georgia Institute of Technology

{yichen.wang, apurvverma, evangelos.theodorou}@gatech.edu, lsong@cc.gatech.edu

Abstract

Our opinion is influenced by those of others we connect to in social networks. How to model opinion dynamics across social networks and can we steer the public opinion to a desired state? Answers to these questions are critically important for understanding the vulnerabilities of online social networks and increasing their resilience to rumor and false information. Recently there has been much work on modeling the dynamics in information diffusion, but few work has integrated these models into a systematic control framework. In this paper, we propose a unified multivariate jump diffusion process framework for modeling opinion dynamics over networks and determining the control over such networks. Our method can handle noisy and partially observed data, and over networks with time-varying and node birth processes. Using synthetic and real world networks, we showed that our framework is robust, able to control both stable and unstable dynamics systems with fast convergence speed, less variance and low control cost.

1 Introduction

Online social networking services have brought to the public a new style of social lives parallel to the day-to-day offline activities. It is a popular trend for people to use social network sites, such as Facebook and Twitter to express their opinions and influence their peers. Consequently, there are many work focusing on predicting people’s opinion (Hegselmann & Krause, 2002; Wang & Pal, 2015), and modeling people’s opinions over time (Holme & Newman, 2006; Das et al., 2014; De et al., 2015). We define this task as modeling the *opinion dynamics*.

Besides modeling the opinion dynamics, there is a growing trend that organizations, such as political campaigns and business merchants, use social media and social networks to influence and steer people’s opinions. For example, government seeks to control the spreading of terrorists’ anti-government opinions, rumors and misinformation over social networks and prevent them from achieving dubious purposes. Moreover, large corporations may pay some influential users to post positive reviews of the products and influence others to have same positive opinion. This leads to the task of *opinion control*, where the decision maker needs to design the best control policy to determine the required level of external drive that should be provided to each user in the social network, such that the opinions of each user can be influenced towards a target state. Hence an ideal opinion model should be able to capture the dynamics accurately and also be flexible to control the dynamics.

However, the majority of the existing studies focuses more on modeling *opinion dynamics*, without including the *opinion control* aspect. The control theoretic point of view of social dynamics is essential for the understanding of the underlying computational principles of such phenomena. Furthermore, answers to these questions are critically important for the understanding the vulnerabilities of online social networks and increasing their resilience to the spread of rumor and false information. In this paper we will propose a framework that is grounded on the theory of stochastic control of Jump Diffusion Processes. More specifically,

- We formulate the opinion dynamics in terms of Stochastic Differential Equations (SDEs) driven by Hawkes processes and Wiener processes, which captures the information dynamics and is robust to noisy data and stochastic disturbances. While there have been previous probabilistic and point process models for opinion dynamics (De et al., 2015), these previous models are not formulated in a SDE framework. Our current formulation and generalization are critical for exploiting the stochastic control framework.
- We propose an efficient stochastic control framework which can steer both stable and unstable opinion dynamics models towards a target state while reducing the variance of the state. The proposed control framework can also deal with the challenging cases of partially observed data and on time-varying networks.
- From a methodological stand point, we extend the stochastic optimal control theory and derive the Ito’s lemma, Bellman’s optimality condition and HJB equation for SDEs driven by multi-dimension temporal point processes (Hawkes processes). In particular, we derive a set of backward Riccati equations that back propagate the value function of the opinion dynamics.
- Our work builds a new bridge between social information diffusion and stochastic control theory for the first time. This opens a new avenue to address the challenging social opinion shaping and policy making problems.

Related work. Most existing work (Clifford & Sudbury, 1973; DeGroot, 1974; Hegselmann & Krause, 2002; Holme & Newman, 2006) consider opinions to be updated sequentially in discrete time. Recently, De et al. (2015) proposed a point process modeling framework which models both the effect of social influence and information diffusion on the opinion. Inspired by this work, our opinion dynamics model not only consider these effects, but also provide a principled stochastic differential equation (SDE) framework for modeling these effects which allow us to consider other additional factors such as baseline tendencies and are more robust to the noise in data. In addition, De et al. (2015) considered steady state behavior of stable systems, while we are providing a unified stochastic control framework for the dynamics for both stable and unstable systems. Furthermore, we showed that our control framework leads to algorithms with faster convergence speed and less variance. In terms of the literature in stochastic optimal control, the jump diffusion SDEs, (Hanson, 2007) is closely related to our model. However, most of the literature (Oksendal & Sulem, 2005; Hanson, 2007; Hu & Oksendal, 2008) focuses on controlling the SDEs with the jump term driven by Poisson processes or compound Poisson processes. Hence significant generalizations are needed for the Hawkes processes and temporal point processes setting.

2 Background on Temporal Point Processes

This section provides necessary concepts of temporal point process (Cox & Isham, 1980; Cox & Lewis, 2006; Daley & Vere-Jones, 2007; Aalen et al., 2008; Du et al., 2015). A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. Many different types of data produced in online social networks can be represented as temporal point processes, such as the times of tweets and link creations. A temporal point process can be equivalently represented as a counting process, $N(t)$, which records the number of events before time t .

An important way to characterize temporal point processes is via the conditional intensity function — a stochastic model for the time of the next event given all the times of previous events. Let $\mathcal{H}(t) = \{t_i | t_i < t\}$ be the history of events happened up to but not including t . Formally, the conditional intensity function $\lambda(t)$ is the conditional probability of observing an event in a small window $[t, t + dt)$ given the history $\mathcal{H}(t)$, *i.e.*,

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

where one typically assumes that only one event can happen in a small window of size dt , *i.e.*, $dN(t) \in \{0, 1\}$. Then, given a time $t' \geq t$, we can also characterize the conditional probability that no event happens during $[t, t')$ and the conditional density that an event occurs at time t' as $S(t') = \exp(-\int_t^{t'} \lambda(\tau) d\tau)$ and

$f(t') = \lambda(t') S(t')$ respectively. Furthermore, we can express the likelihood of a list of events $\{t_i\}_{i=1}^n$ in an observation window $[0, T)$ as

$$\ell(\{t_i\}_{i=1}^n) = \prod_{i=1}^n \lambda(t_i) \exp\left(-\int_0^T \lambda(\tau) d\tau\right) \quad (1)$$

This simple likelihood will later enable us to learn the parameters from observed data.

3 Opinion Control Model

In this section, we will introduce the opinion dynamics in the form of a jump diffusion Stochastic Differential Equation (SDE). The model is inspired by the point process model of (De et al., 2015). However, we make significant generalization using the SDE framework, and furthermore without this generalization, it is not easy to design stochastic control algorithms which we will present later.

More specifically, we use $\mathbf{x}(t) \in \mathbb{R}^U$ to represent the vector of each people's opinion in the social network with U users. For each user v , $\mathbf{x}_v = 0$ stands for neutral opinion and large positive/negative denotes corresponding opinion. We consider the following SDE to model the dynamics:

$$d\mathbf{x}(t) = \underbrace{(\mathbf{b} - \mathbf{x}(t) + \theta \mathbf{u})dt}_{\text{drift}} + \underbrace{\beta d\mathbf{w}(t)}_{\text{diffusion}} + \underbrace{\mathbf{h}(\mathbf{x}(t))d\mathbf{N}(t)}_{\text{jump}} \quad (2)$$

where $\mathbf{w}(t) \in \mathbb{R}^U$ is the Wiener process capturing the noise, and $\mathbf{N}(t) \in \mathbb{R}^U$ is the Hawkes Process, which captures the influence of information propagation on ones' opinion. The parameters β, θ are nonnegative constants. $\mathbf{u}(t) \in \mathbb{R}^U$ is the control process. $\mathbf{h}(\mathbf{x}) : \mathbb{R}^U \rightarrow \mathbb{R}^{U \times U}$ is a matrix with $\mathbf{h}(\mathbf{x})_{ij} = \mathbf{M}_{ij}\mathbf{x}_j$. Mathematically,

$$\mathbf{h}(\mathbf{x}) = \mathbf{M} \odot \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_U \\ \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \cdots & \mathbf{x}_U \end{pmatrix} \quad (3)$$

where $\odot$ is the element-wise product. $\mathbf{M} \in \mathbb{R}^{U \times U}$ is a given weighted network adjacency matrix, and $\mathbf{M}(u, v)$ denotes the influence of u 's opinion v . Note $\mathbf{M}(u, v) = 0$ means u and v are not connected in the network. Next, we discuss the intuition behind each term in detail.

The drift process. The opinion's change rate, $d\mathbf{x}(t)/dt$, is negative proportional to its opinion $\mathbf{x}(t)$. This is because people's opinion tend to stabilize over time. $\mathbf{b}$ is the baseline opinion of each user, *i.e.*, personal characteristics. Without the control and the jump term, the expected value of $\mathbf{x}(t)$ will move to $\mathbf{b}$ as time goes by. This can be seen by taking expectation of the dynamics and setting $d\mathbb{E}[\mathbf{x}] = \mathbf{0}$.

The control process. $\mathbf{u}(\mathbf{x}, t)$ is a feedback control, which means it depends on the feedback opinion $\mathbf{x}(t)$. We control the opinion network on the drift term and $\mathbf{u}$ determines how fast the opinion needs to be changed. For example, business merchants may pay the customer to change his review of a product from -3 to 1 in one day and the control cost (discussed later) is the price to pay the customer.

The diffusion process. This term captures the normal fluctuations in the dynamic system. Besides the social network's influence, other factors will also influence a user's opinion, such as personality, the news he reads everyday, and other unexpected things. This change can be described by Wiener process $\mathbf{w}(t)$. Two key property of $\mathbf{w}(t)$ are $\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, t - s)$ and it has independent increments. $\mathcal{N}(0, \sigma)$ denotes the Gaussian distribution with zero mean and standard deviation σ . This term makes the model much more robust to the noise.

The jump process. This term captures abrupt changes in opinions, due to the arrival of new information from other nodes in the social networks. Inspired by the idea in (De et al., 2015), we model the *social influence* and *information dynamics*, but in the form of SDEs. First, for social influence, we use $d\mathbf{N}(t)$ to denote the number of message events per user in the window $[t, t + dt)$ and the coefficient $\mathbf{h}(\mathbf{x})$ of $d\mathbf{N}(t)$ captures the positive or negative effects of the social influence. There are two types of influence: mutual excitation and self excitation. In the case of mutual excitation, if user u 's friend v has tweeted about something bad in his life and has negative opinion, *i.e.*, $\mathbf{x}_v(t)$ is negative, then the u ' opinion $\mathbf{x}_u(t)$ is likely to change towards negative. Moreover, in the case of self excitation, if u himself tweets positively, $\mathbf{x}_u(t)$ is inclined to move

towards positive. Mathematically, we use $\sum_{v:\mathbf{M}(u,v)=1} \mathbf{x}_v d\mathbf{N}_v$ to capture the sum of u 's neighbors' opinion influence (including himself). It is equivalent to the jump term in u -th dimension in (2).

Moreover, for the *information dynamics*, we use the multidimensional Hawkes process (Liniger, 2009; Farajtabar et al., 2014) to model $\mathbf{N}(t)$. We capture the mutual excitation behavior of tweeting/retweeting events by assuming the intensity of such event of a user is boosted by previous events from his neighbors. The intensity $\lambda_u(t)$ for the user u is expressed as:

$$\lambda_u(t) = \eta_u + \sum_{i:t_i < t} \alpha_{uu_i} \kappa(t - t_i) \quad (4)$$

where $\kappa(t) = \exp(-t)$ is an exponential triggering kernel, $\eta_u \geq 0$ is a baseline intensity and models the publication of message on the user's own initiative. The second term $\alpha_{uu'} \geq 0$ models the publication of additional messages by u due to the social influence. Here, the occurrence of each historical event increases the intensity by a certain amount determined by the triggering kernel $\kappa(t - t_i)$ and the weight $\alpha_{uu'}$. We will focus on the exponential kernel in this paper. However, other functional forms for the triggering kernel, such as log-logistic function, are possible, and our model does not depend on this particular choice.

Remark. Compared with (De et al., 2015), our SDE formulation of the opinion dynamics also considers the social influence on one's opinion and models the information dynamics using Hawkes processes. However, there are two major differences: i) we not only consider the social influence, but also have a Wiener diffusion part that considers other baseline factors that influence one's opinion, which makes the model more robust to noisy data. ii) The formulation in (De et al., 2015) is not easy to be connected to the stochastic optimal control theory. Our SDE view is novel and provides a natural connection. Hence, we can easily conduct stochastic optimal control over the dynamics.

Given the opinion dynamics, a nature question is how should one control over it and steer it towards the target state? Next we formalize the control framework.

4 Opinion Control Formulation

With the opinion control model, we now present the optimal control framework for the opinion dynamics. We will provide an overview of solutions to the optimal control problem near the end of this section. More specifically, the control objective function has the form:

$$V(\mathbf{x}(t), t) = \min_{\mathbf{u}} \mathbb{E} \left[\int_t^T \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}, \tau) d\tau + \phi(\mathbf{x}(T), T) \right]$$

subject to the final condition: $V(\mathbf{x}(T), T) = \phi(\mathbf{x}(T), T)$. Here t and T are the start and terminal time respectively. V is called the value function. $\mathcal{L}: \mathbb{R}^U \times \mathbb{R}^U \times \mathbb{R} \rightarrow \mathbb{R}^+$ is the path cost and $\phi: \mathbb{R}^U \times \mathbb{R} \rightarrow \mathbb{R}^+$ is the terminal cost. They are utilities of the network. The expectation of total cost accumulated from t_0 to T is taken over the joint stochastic pair $\{\mathbf{w}(t), \mathbf{N}(t)\}$ for $t \in (t_0, T]$. The goal is to find the optimal control $\mathbf{u}^*$ s.t. the minimization is reached. We use the terminal cost q and path cost $\mathcal{L}$ as follows:

$$\phi(\mathbf{x}(T), T) = \frac{1}{2} (\mathbf{x}(T) - \mathbf{a})^\top (\mathbf{x}(T) - \mathbf{a}) \quad (5)$$

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, t) = \underbrace{\frac{1}{2} (\mathbf{x}(t) - \mathbf{a})^\top \mathbf{Q} (\mathbf{x}(t) - \mathbf{a})}_{\text{tracking error}} + \underbrace{\frac{1}{2} \mathbf{u}^\top \mathbf{R} \mathbf{u}}_{\text{control cost}} \quad (6)$$

where $\mathbf{a} \in \mathbb{R}^U$ is a target state vector, $\mathbf{Q} \in \mathbb{R}^{U \times U}$ and $\mathbf{R} \in \mathbb{R}^{U \times U}$ are given coefficients. Next, we discuss the intuition behind the formulations.

Terminal cost ϕ . The control objective is to make the expected opinion in the social network to achieve some ideal state $\mathbf{a}$, i.e., everyone is positive at the final time T . Mathematically, with $\mathbb{E}[\mathbf{x}(T)] = \mathbf{a}$, we aim to minimize the variance $\mathbb{E} \|\mathbf{x}(T) - \mathbf{a}\|_F^2$.

Path cost $\mathcal{L}$. At each time $t \in (t_0, T]$, we control $\mathbf{x}(t)$ to have the desired trajectory of $\mathbf{a}$. Similar to the terminal cost, we define the tracking error as $(\mathbf{x} - \mathbf{a})^\top \mathbf{Q} (\mathbf{x} - \mathbf{a})$. Moreover, at each time, we minimize the control cost as $\mathbf{u}^\top \mathbf{R} \mathbf{u}$, which denotes how much effort of the control $\mathbf{u}$ is put into the dynamics. If $\mathbf{u}$ is near $\mathbf{0}$, then the control cost is low. Generally, if there is no coupling information between each dimension of $\mathbf{x}$ and $\mathbf{u}$, we set $\mathbf{Q}$ and $\mathbf{R}$ to be diagonal. We next provide two cases to choose $\mathbf{Q}$ and $\mathbf{R}$.

- $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \rho \mathbf{I}$. The parameter ρ controls the trade off between the control cost and the tracking error. We will use this parametrization in our model.
- $\mathbf{Q} = \text{diag}(q_1, \dots, q_U)$ and $\mathbf{R} = \text{diag}(r_1, \dots, r_U)$. We can assign different weights to each dimension, which is particularly useful if we know some users are more influential than others, such as the leader in the society and the person who has many connections. Assigning higher weights can give high priority in controlling them.

To solve the optimal control problem, in section 5, we first present the framework to solve the **continuous and fully observed control**, which is common in the robotics and aerospace applications. In theory, this is the *best* control if i) we have continuous and full observation of the state, and ii) can control the network at any time t . However, these two assumptions are unrealistic. Hence, built upon this method, in section 6, we present the **piecewise-constant control**, which i) efficiently handles **partially observed data**, and ii) controls the network whenever an event happens and is kept constant until next event happens. Finally, in section 7, we extend our control framework to the case of **time-varying and size-increasing** network and present the min-max robust control framework.

5 Continuous and Fully Observed Control

In this section we solve the continuous and fully observed control problem, which falls into the category of the Linear Quadratic (LQ) control. The LQ problem is linear in the dynamics and quadratic in the costs with respect to both the state $\mathbf{x}$ and the control $\mathbf{u}$. However, prior work all focus on the SDE where the jump term is driven by a Poisson process (Westman & Hanson, 1997; Oksendal & Sulem, 2005; Hanson, 2007) and significant generalizations are needed for our setting. This is due to the special structure of the opinion dynamics, *i.e.*, the jump term $d\mathbf{N}(t)$ in (2) is driven by a Hawkes process and the weights $\mathbf{h}(\mathbf{x})$ considers the social influence. We are the first to derive algorithms for this structure.

Bellman's optimality. To obtain the optimal control, we provide the dynamic programming framework. Bellman's optimality has been studied extensively SDEs driven by Poisson and Wiener processes in (Yong & Zhou, 1999; Hanson, 2007). Here we extend to general point processes.

Theorem 1. *Bellman's Optimality condition yields*

$$V(\mathbf{x}(t), t) = \min_{\mathbf{u}} \mathbb{E}_{(\mathbf{w}, \mathbf{N})(t+\Delta t)} \left\{ \int_t^{t+\Delta t} \mathcal{L}(\mathbf{x}(t), \mathbf{u}, s) ds + V(\mathbf{x}(t + \Delta t), t + \Delta t) \right\} \quad (7)$$

Speaking intuitively, the expression in brackets represents the sum of the running cost on $[t, t + \Delta t]$ and the minimum cost obtained by proceeding optimally on $[t + \Delta t, T]$ with $\mathbf{x}(t + \Delta t)$ as the initial state.

Ito's formula. The Ito's lemma (Hanson, 2007) designed for Poisson driven SDE plays a key role in the stochastic calculus. Here we generalize it to the multi-dimension and general point process.

Theorem 2 (Generalized Itô Formula). *Assume $\mathbf{x}(t)$ has dynamics of the form:*

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{g}(\mathbf{x}, t)d\mathbf{w}(t) + \mathbf{h}(\mathbf{x}, t)d\mathbf{N}(t)$$

Let $V(\mathbf{x}, t)$ be a twice-differentiable scalar function function in $\mathbf{x}$ and once in t . Then the formula holds:

$$dV(\mathbf{x}(t), t) = \left\{ V_t + \frac{1}{2} \text{tr}(V_{\mathbf{x}\mathbf{x}} \mathbf{g} \mathbf{g}^\top) + V_{\mathbf{x}}^\top \mathbf{f} \right\} dt + V_{\mathbf{x}}^\top \mathbf{g} d\mathbf{w}(t) + (V(\mathbf{x} + \mathbf{h}, t) - V(\mathbf{x}, t)) d\mathbf{N}(t)$$

This formula can be seen as the stochastic version of a Taylor expansion of $V(\mathbf{x}, t)$ stopped at the second order.

Combining Bellman's optimality and the Itô's formula, we derive the Hamilton-Jacobi-Bellman (HJB) equation. Solving the HJB equation leads to the optimal control.

Hamilton-Jacobi-Bellman equation. Using the Bellman optimality, generalized Ito's formula with V as the value function, and by taking the limit of $\Delta t \rightarrow 0$, replacing Δt by dt in (7), we can systemically

derive HJB equation as:

$$-\frac{\partial V}{\partial t} = \min_{\mathbf{u}} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \frac{\beta^2}{2} \text{tr}(V_{\mathbf{x}\mathbf{x}}(\mathbf{x}, t)) + V_{\mathbf{x}}(\mathbf{x}, t)^\top (\mathbf{b} - \mathbf{x}(t) + \theta \mathbf{u}) + \sum_{j=1}^U \lambda_j(t) \left(V(\mathbf{x} + \mathbf{h}_j(\mathbf{x}), t) - V(\mathbf{x}, t) \right) \right\} \quad (8)$$

where $\lambda(t)$ is the intensity for the multidimensional Hawkes process defined in (4). This HJB equation is subject to the final condition: $V(\mathbf{x}(T), T) = \phi(\mathbf{x}(T), T)$.

Note that $\mathbf{h}_j(\mathbf{x}) \in \mathbb{R}^U$ in (8) corresponds to the j^{th} column of $\mathbf{h}$ and is the jump amplitude vector of user j . That is, how much influence j 's opinion has on other users. Since j will only influence his friends, we have $\mathbf{h}_j(v) = 0$ if $\mathbf{M}(j, v) = 0$. Hence we have:

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{M}(:, j) \odot (\mathbf{x}_j, \dots, \mathbf{x}_j)^\top = \mathbf{A}_j \mathbf{x} \quad (9)$$

where $\mathbf{A}_j$ is a $U \times U$ matrix with the j^{th} column to be $\mathbf{A}_j(:, j) = \mathbf{M}(:, j)$ and 0 elsewhere.

To solve the HJB equation, we can assume that the value function V is quadratic in $\mathbf{x}(t)$ with unknown coefficients $\mathbf{v}_1(t) \in \mathbb{R}^U$, $\mathbf{v}_{11}(t) \in \mathbb{R}^{U \times U}$ and $v_0(t) \in \mathbb{R}$:

$$V(\mathbf{x}(t), t) = v_0(t) + \mathbf{v}_1(t)^\top \mathbf{x}(t) + \frac{1}{2} \mathbf{x}(t)^\top \mathbf{v}_{11}(t) \mathbf{x}(t)$$

Note that this assumption only works for the Linear-Quadratic problem due to the special structure of the dynamics and the cost (Hanson, 2007).

To find the optimal control, we first substitute $V(\mathbf{x}, t)$ to HJB equation (8) and take the gradient of the right side of (8) with respect to $\mathbf{u}$ and set it to $\mathbf{0}$. This yields the optimal control:

$$\mathbf{u}^* = -\frac{\theta}{\rho} V_{\mathbf{x}} = -\frac{\theta}{\rho} (\mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{x}(t)) \quad (10)$$

We can see that the optimal control policy consists of a feedforward and feedback term which is linear in the opinion state. The remaining step is to find $\mathbf{v}_1(t)$, $\mathbf{v}_{11}(t)$ and $v_0(t)$ to obtain V and $\mathbf{u}^*$. We begin with substituting $\mathbf{u}^*$ back to the HJB equation in (8), and separate it into purely quadratic in $\mathbf{x}$, purely linear and state-independent terms. See appendix for the complete derivation. This leads to a set of three uni-directionally coupled matrix Ordinary Differential Equations (ODEs):

Update $\mathbf{v}_{11}(t)$. We solve the $U \times U$ quadratic coefficient equation with terminal condition $\mathbf{v}_{11}(T) = \mathbf{I}$:

$$-\frac{d\mathbf{v}_{11}(t)}{dt} = \mathbf{I} + 2\mathbf{v}_{11}(t)(-1 + \mathbf{\Lambda}_1(t)) + \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j^\top \mathbf{v}_{11}(t) \mathbf{A}_j - \frac{\theta^2}{\rho} \mathbf{v}_{11}(t) \mathbf{v}_{11}(t) \quad (11)$$

where $\mathbf{\Lambda}_1(t) = \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j$.

Update $\mathbf{v}_1(t)$. We solve the $U \times 1$ linear coefficient equation with $\mathbf{v}_1(T) = -\mathbf{a}$:

$$-\frac{d\mathbf{v}_1(t)}{dt} = -\mathbf{a} + (-1 + \mathbf{\Lambda}_1^\top(t) - \frac{\theta^2}{\rho} \mathbf{v}_{11}(t)) \mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{b} \quad (12)$$

Update $v_0(t)$. We solve the scalar ODE with $v_0(T) = 0$:

$$-\frac{dv_0(t)}{dt} = \mathbf{b}^\top \mathbf{v}_1(t) + \frac{\beta^2}{2} \text{tr}(\mathbf{v}_{11}(t)) - \frac{\theta^2}{2\rho} \mathbf{v}_1^\top(t) \mathbf{v}_1(t) \quad (13)$$

The above three ODEs should be solved in sequential order. $\mathbf{v}_{11}(t)$ is solved first, then substituted into (12) to compute $\mathbf{v}_1(t)$. Finally both $\mathbf{v}_{11}(t)$ and $\mathbf{v}_1(t)$ are substituted to compute $v_0(t)$. The optimal control $\mathbf{u}^*$ and cost V are then obtained. Note that it is sufficient to compute $\mathbf{v}_{11}(t)$, $\mathbf{v}_1(t)$ to obtain $\mathbf{u}^*$. Since $\mathbf{v}_{11}(t)$ is a symmetric matrix by being defined with a quadratic form, only a triangle part of $\mathbf{v}_{11}(t)$ needs to be solved, i.e., just $n(n+1)/2$ elements. Note that for (11), it is similar to the matrix Riccati differential equation (Nazarzadeh et al., 1998). However, unlike prior work (Nguyen & Gajic, 2010) for the SDE dynamics without jump process, it is not possible to get an analytical solution for $\mathbf{v}_{11}(t)$ due to the term $\sum_{j=1}^U \lambda_j \mathbf{A}_j^\top \mathbf{v}_{11} \mathbf{A}_j$, which comes from the jump term $\mathbf{h}(\mathbf{x}) d\mathbf{N}(t)$.

To solve $\mathbf{v}_{11}(t)$, $\mathbf{v}_1(t)$, we use the numerical Runge-Kutta formula (Dormand & Prince, 1980). We will partition $(t_0, T]$ to equally-spaced time stamps $\{\tau_k\}$ and obtain the values of $\mathbf{v}_{11}(t)$, $\mathbf{v}_1(t)$ at these time

Algorithm 1 ODE-SOLVER

```
1: Input:  $M, \mathbf{a}, \boldsymbol{\eta}, \beta, \mathbf{b}, \{\tau_k\}_{k=1}^m, \{t_i\}_{i=1}^n$ 
2: Output:  $\mathbf{v}_{11}(\tau_k), \mathbf{v}_1(\tau_k), k = 1, \dots, m$ 
3: for  $j = 1$  to  $U$  do
4:   Set matrix  $\mathbf{A}_j(:, j) = M(:, j)$  and 0 elsewhere
5: end for
6: for  $k = 1$  to  $m$  do
7:   Compute  $\boldsymbol{\lambda}_u(\tau_k) = \boldsymbol{\eta}_u + \sum_{j:t_i < \tau_k} \alpha_{uu_i} \kappa(\tau_k - t_i)$ 
8:   Compute  $\boldsymbol{\Lambda}_1(\tau_k) = \sum_{u=1}^U \boldsymbol{\lambda}_u(\tau_k) \mathbf{A}_u$ 
9: end for
10: With  $\{\mathbf{A}_j\}, \boldsymbol{\Lambda}_1$ , compute  $\mathbf{v}_{11}(\tau_k), \mathbf{v}_1(\tau_k)$  in (11) and (12) using ODE45 in MATLAB
```

stamps. Algorithm 1 summarizes the procedure.

Based on the continuous control, we will derive a more practical control which deals with the partially observed data efficiently in the next section.

6 Piecewise-constant and Partially Observed Control

In real world, the observed data is in the form of $\mathcal{T} = \{(t_i, \mathbf{x}_{v_i}, v_i)\}_{i=1}^n$. Each triplet means a user v_i expresses his opinion $\mathbf{x}_{v_i}$ at time t_i . Hence two distinguishing feature of the opinion control problem are i) partial observation, *i.e.*, we only have access to one dimension v_i of $\mathbf{x}$ at time t_i and do not observe $\mathbf{x}$ between events and, ii) piecewise-constant, *i.e.*, we will only control the network whenever some user expresses his opinion and will keep using the same control until new event happens. To tackle the partial observation problem, we propose the time-variant control and then obtain the piecewise-constant control based on it.

6.1 Time-variant Control

First we discuss the *time-variant control* $\mathbf{u}_1$, which has the following structure:

$$\mathbf{u}_1(t) = \mathbf{u}_1(t, \mathbf{x}(t_i)), \quad t_i \leq t < t_{i+1}$$

This control is a function of the time t and $\mathbf{x}(t_i)$ that includes partially-observed state $\mathbf{x}_{v_i}$ and other states from estimation. Set $\mathcal{I}_i = \{\mathbf{x}_{v_i}(t_i), t_i\}$ to be the available information at the controller at time t . Since no further observations of the state $\mathbf{x}(t)$ is available on the interval $[t_i, t_{i+1})$, then motivated by the separation principle (Ades et al., 2000; Molin & Hirche, 2010), we replace $\mathbf{x}(t)$ by its expected value $\mathbb{E}[\mathbf{x}(t)|\mathcal{I}_i]$ with the initial state $\mathbf{x}(t_i)$ known. Then the optimal control is expressed as:

$$\mathbf{u}_1^* = -\frac{\theta}{\rho}(\mathbf{v}_1(t) + \mathbf{v}_{11}(t)\mathbb{E}[\mathbf{x}(t)|\mathcal{I}_i]) \quad (14)$$

Next we show how to compute the estimated state $\mathbb{E}[\mathbf{x}(t)|\mathcal{I}_i]$. We substitute (14) to (2) and take the expectation, then obtain the dynamics of $\mathbb{E}[\mathbf{x}(t)|\mathcal{I}_i]$ on $[t_i, t_{i+1})$:

$$\begin{aligned} d\mathbb{E}[\mathbf{x}|\mathcal{I}_i] &= (\mathbf{b} - \mathbb{E}[\mathbf{x}|\mathcal{I}_i] - \theta^2(\mathbf{v}_1(t) + \mathbf{v}_{11}(t)\mathbb{E}[\mathbf{x}|\mathcal{I}_i])dt + \mathbf{h}(\mathbb{E}[\mathbf{x}|\mathcal{I}_i])\boldsymbol{\lambda}(t)dt \\ \mathbb{E}[\mathbf{x}_{v_i}(t_i)|\mathcal{I}_i] &= \mathbf{x}_{v_i}(t_i) \end{aligned} \quad (15)$$

Now we derive the method to compute $\mathbf{u}_1^*$: on each interval $[t_i, t_{i+1})$, with observation $\mathbf{x}_{v_i}(t_i)$, we solve (15) to obtain the estimation $\mathbb{E}[\mathbf{x}(t)|\mathcal{I}_i]$ for $t_i < t < t_{i+1}$, then we obtain the optimal control $\mathbf{u}_1^*$ using (14).

6.2 Piecewise-constant Control

Although the time-variant control $\mathbf{u}_1^*$ deals with partial observation efficiently, it still relies on the assumption that controller can control the network at anytime, which might not be able to be executed easily in practice. In reality, the controller will not control the opinion dynamics constantly and only control the network whenever some user expresses his opinion and will keep using the same control until new event happens.

Algorithm 2 PIECEWISE-CONSTANT CONTROL

```

1: Input: Params  $\mathcal{P} = \{\alpha, \eta, \beta, \mathbf{b}, \theta, \mathbf{M}, T\}, \{t_i\}_{i=1}^n$ 
2: Output: Piecewise-constant control  $\mathbf{u}_2^*$ 
3: Discretize  $(0, T]$  with  $\{\tau_k\}_{k=1}^m$  s.t.  $\tau_{k+1} - \tau_k = 0.1$ ,  $\tau_1 = 0$  and  $\tau_m = T$ ,  $m = 10T + 1$ 
4: for  $k = 1$  to  $m$  do
5:    $\mathbf{v}_{11}(\tau_k), \mathbf{v}_1(\tau_k) = \text{ODE-SOLVER}(\mathcal{P}, \{t_i\}, \{\tau_k\})$ 
6: end for
7: for  $k = 1$  to  $m$  do
8:   Compute  $\mathbb{E}[\mathbf{x}(\tau_k)]$  using (15)
9: end for
10: for  $i = 1$  to  $n$  do
11:   Compute  $\mathbf{u}_1^*(\tau_k) = -\frac{\theta}{\rho}(\mathbf{v}_1(\tau_k) + \mathbf{v}_{11}(\tau_k)\mathbb{E}[\mathbf{x}(\tau_k)])$  for  $\tau_k \in [t_i, t_{i+1})$ 
12:   Compute  $\lambda_u(t_i) = \eta_u + \sum_{j: t_j < t_i} \alpha_{uu_i} \kappa(t_i - t_j)$ 
13:   Compute  $\delta t = \min_{u \in U} \frac{1}{\lambda_u(t_i)}$ 
14:   Compute  $\mathbf{u}_2^*(t) = \text{Average}\{\mathbf{u}_1^*(\tau_k) | \tau_k \in [t_i, t_i + \delta t)\}$  for  $t \in [t_i, t_{i+1})$ 
15: end for

```

Hence the second type control: *piecewise constant* control is more realistic. It is constant on each interval and a function of the opinion on the start of the interval:

$$\mathbf{u}_2(t) = \mathbf{u}_2(\mathbf{x}(t_i)), \quad t_i \leq t < t_{i+1}$$

We propose an efficient algorithm to compute optimal piecewise constant control. After event t_i , the process can be treated as the Poisson process with intensity $\lambda(t_i)$, hence we can obtain the expected waiting time until next event as $\delta t = \min_{v \in U} \frac{1}{\lambda_v}$ (Aalen et al., 2008). Then on the interval $[t_i, t_{i+1})$, we obtain the optimal control as the average of $\mathbf{u}_1^*$ on the interval $[t_i, t_i + \delta t)$:

$$\mathbf{u}_2^*(t) = \text{Average}\{\mathbf{u}_1^*(t) | t \in [t_i, t_i + \delta t)\} \quad (16)$$

In the experiments, we show that this approximation has comparable performance with the continuous fully observed control in (10). Algorithm 2 summarizes the idea.

7 Extensions

Here we first extend our control framework to time-varying network and node birth network. Finally we present the **first** framework for min-max robust control for Jump Diffusion SDEs.

7.1 Control with Time-varying Network

In reality the social network changes over time and is not fixed (Farajtabar et al., 2015). Users can follow or unfollow each other as time goes by. Our control framework can be easily extended to the time-varying network case. Note that for the control with fixed network, the conditional expectation is over $\{\mathbf{w}(t), \mathbf{N}(t)\}$ when using Bellman's optimality to derive the HJB equation. However, now we also need to take the expectation over the adjacency matrix $\mathbf{M}(t)$. We modify the HJB equation in (8) and take expectation over the jump term $\mathbf{h}_i(\mathbf{x})$:

$$\sum_{j=1}^U \lambda_j(t) (V(\mathbf{x} + \mathbb{E}[\mathbf{h}_j(\mathbf{x})], t) - V(\mathbf{x}, t)) \quad (17)$$

where $\mathbb{E}[\mathbf{h}_j(\mathbf{x})] = \mathbb{E}[\mathbf{M}(:, j)] \odot (\mathbf{x}_j, \dots, \mathbf{x}_j)^\top$. Hence we will use $\mathbb{E}[\mathbf{M}(t)]$ instead of $\mathbf{M}$ as input to Algorithm 2 to compute the optimal control. Next, we show how to compute $\mathbb{E}[\mathbf{M}(t)]$ for the link creation process. Suppose we know the total number of nodes U at the terminal time T . Then we set $\mathbf{M}(0) = \mathbf{I}$. We model the intensity, $\sigma(t)$, link creation using a combination of survival and Poisson process:

$$\sigma_{vs}(t) = (1 - \mathbf{M}_{vs}(t)) \gamma_v \quad (18)$$

where the term $1 - \mathbf{M}_{vs}(t)$ effectively ensures a link is created only once, and the intensity is set to zero after that.

The term $\gamma_v \geq 0$ denotes the Poisson intensity, which models when a node v decides to follow another node spontaneously at his own initiative. Since the link creation process is a special case of the temporal point process, from (Aalen et al., 2008), we have

$$d\mathbb{E}[\mathbf{M}_{vs}(t)] = \sigma_{vs}(t)dt = (1 - \mathbb{E}[\mathbf{M}_{vs}(t)])\gamma_v dt \quad (19)$$

with initial condition $\mathbb{E}[\mathbf{M}_{vs}(0)] = 0$ if s and v are not connected. We can easily solve the first order matrix differential equation (19) and obtain $\mathbb{E}[\mathbf{M}_{vs}(t)]$ as:

$$\mathbb{E}[\mathbf{M}_{vs}(t)] = 1 - \exp(-\gamma_v t)$$

7.2 Control with Node Birth Process

The network dimension can grow as time goes by since more and more people will join the network. How to control the dynamics in this case remains unknown. Here we provide an efficient and novel way to handle this problem by establishing the connection to the time-varying network case. More specifically, we have the following observation.

Observation. *The process of adding a new user v to the existing network $\mathbf{M} \in \mathbb{R}^{(N-1) \times (N-1)}$ and connects to user s is equivalent to link creation process of setting $\mathbf{M}(t) \in \mathbb{R}^{N \times N}$ to be the existing network and letting $\mathbf{M}(v, s) = 1$.*

Instead of having a sequence of size-growing matrix $\mathbf{M}(t)$, a memory efficient way is to fix its dimension to be the maximum number of nodes at terminal time and adds a link whenever a user joins the network. Moreover, it enables us to transform the stochasticity of node birth process, which changes the network's dimension, to the stochasticity of link creation process with fixed network dimension. It remains unknown how to derive HJB equation when the network's dimension is not fixed.

Hence the problem boils down to the time varying network case. Finally, a key difference between the controls for time-varying network and node birth network is: for the former one, every node is in the network and the optimal control is on every node. For the node birth case, we will set the control of this node to be 0 until it is added.

7.3 Min-max Control

Here, we extend our framework to the case of min-max control. The min-max control has been studied for the SDEs with only the diffusion part (Morimoto & Atkeson, 2003). However, there is **no** work on the area for Jump Diffusion SDEs and we provide the **first** min-max control framework. In the min-max control framework, we have two opponents. Besides the control $\mathbf{u}$, there is also a disturbance $\mathbf{v}$ in the system that tries to destabilize the system and maximize the cost. Now we try to seek a stabilizing control $\mathbf{u}$ that minimize the cost for the worst disturbance $\mathbf{v}$. Hence the stochastic optimal control problem can be formulated as a **differential game** with two opponents $\mathbf{u}, \mathbf{v}$.

The min-max control is particular useful for the scenarios where the control $\mathbf{u}$ is competing with the disturbance $\mathbf{v}$. For example, in social network sites, the terrorists try to create disturbance $\mathbf{v}$ and spread anti-government opinions and rumors. The government will provide external drive $\mathbf{u}$ to compete against the disturbance, which leads to the min-max control problem.

The dynamics is similar to (2) with an extra disturbance $\mathbf{v}$:

$$d\mathbf{x}(t) = \underbrace{(\mathbf{b} - \mathbf{x}(t) + \theta\mathbf{u} + \theta_2\mathbf{v})dt}_{\text{drift}} + \underbrace{\beta d\mathbf{w}(t)}_{\text{diffusion}} + \underbrace{\mathbf{h}(\mathbf{x})d\mathbf{N}(t)}_{\text{jump}} \quad (20)$$

The objective of min-max control is to seek $\mathbf{u}$ to minimize the cost with the worst disturbance $\mathbf{v}$.

$$V(\mathbf{x}(t), t) = \min_{\mathbf{u}} \max_{\mathbf{v}} \mathbb{E} \left[\int_t^T \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}, \mathbf{v}, \tau) d\tau + \phi(\mathbf{x}(T), T) \right]$$

where $\mathbf{u}$ minimizes the cost function while the destabilizing control $\mathbf{v}$ maximizes it. $\mathcal{L}$ is defined as:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}, t) = \frac{1}{2}(\mathbf{x}(t) - \mathbf{a})^\top (\mathbf{x}(t) - \mathbf{a}) + \frac{\rho}{2}\mathbf{u}^\top \mathbf{u} - \frac{\gamma}{2}\mathbf{v}^\top \mathbf{v} \quad (21)$$

where $\gamma, \rho > 0$.

To solve the min-max control problem, similar to (8) we can derive the HJB equation as follows:

$$\begin{aligned} -V_t = \min_{\mathbf{u}} \max_{\mathbf{v}} & \left\{ \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}, t) + \frac{\beta^2}{2} \text{tr}(V_{\mathbf{x}\mathbf{x}}(\mathbf{x}, t)) + V_{\mathbf{x}}(\mathbf{x}, t)^\top (\mathbf{b} - \mathbf{x}(t) + \theta \mathbf{u} + \theta_2 \mathbf{v}) \right. \\ & \left. + \sum_{j=1}^U \lambda_j(t) \left(V(\mathbf{x} + \mathbf{h}_j(\mathbf{x}), t) - V(\mathbf{x}, t) \right) \right\} \end{aligned} \quad (22)$$

Taking gradient of (22) with respect to $\mathbf{u}$ and $\mathbf{v}$ respectively, we can obtain the optimal control $\mathbf{u}^*$ and the worst disturbance $\mathbf{v}^*$ (destabilizing controller) as:

$$\mathbf{u}^* = -\frac{\theta}{\rho} V_{\mathbf{x}} = -\frac{\theta}{\rho} (\mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{x}(t)) \quad (23)$$

$$\mathbf{v}^* = \frac{\theta_2}{\gamma} V_{\mathbf{x}} = \frac{\theta_2}{\gamma} (\mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{x}(t)) \quad (24)$$

where V is defined the same as that in (5). We can see that for $\gamma \rightarrow \infty$, $\mathbf{v}^* \rightarrow 0$ and the min-max control problem reduces to the control in Section 5.

Now we substitute $\mathbf{u}^*$ and $\mathbf{v}^*$ back to the HJB equation in (22). Set $\zeta = (\frac{\theta^2}{2\rho} - \frac{\theta_2^2}{2\gamma})$ and we also obtain three ODEs as:

Update $\mathbf{v}_{11}(t)$. We solve the $U \times U$ quadratic coefficient equation with terminal condition $\mathbf{v}_{11}(T) = \mathbf{I}$:

$$-\frac{d\mathbf{v}_{11}(t)}{dt} = \mathbf{I} + 2\mathbf{v}_{11}(t)(-1 + \mathbf{\Lambda}_1(t)) + \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j^\top \mathbf{v}_{11}(t) \mathbf{A}_j - 2\zeta \mathbf{v}_{11}(t) \mathbf{v}_{11}(t) \quad (25)$$

where $\mathbf{\Lambda}_1(t) = \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j$.

Update $\mathbf{v}_1(t)$. We solve the $U \times 1$ linear coefficient equation with $\mathbf{v}_1(T) = -\mathbf{a}$:

$$-\frac{d\mathbf{v}_1(t)}{dt} = -\mathbf{a} + (-1 + \mathbf{\Lambda}_1^\top(t) - 2\zeta \mathbf{v}_{11}(t)) \mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{b} \quad (26)$$

Update $v_0(t)$. We solve the scalar ODE with $v_0(T) = 0$:

$$-\frac{dv_0(t)}{dt} = \mathbf{b}^\top \mathbf{v}_1(t) + \frac{\beta^2}{2} \text{tr}(\mathbf{v}_{11}(t)) - \zeta \mathbf{v}_1^\top(t) \mathbf{v}_1(t) \quad (27)$$

Compare the ODEs in (25,26,27) with these in (11,12,13), the only difference is that $\zeta = \frac{\theta^2}{2\rho}$ in (11,12,13). Hence We can use Algorithm 1 to solve these ODEs and obtain $\mathbf{u}^*, \mathbf{v}^*$. Moreover, all the above discussions for piecewise-constant control, time-varying network and size-increasing network can be applied for the min-max control.

8 Parameter Estimation

To control the opinion dynamics, we first need to learn the parameters of the uncontrolled dynamics in (2). There are two sets of parameters: i) $\{\mathbf{b}, \beta\}$ are the coefficients of drift and diffusion processes, and ii) $\{\boldsymbol{\eta}, \boldsymbol{\alpha}\}$ are the coefficients of the Hawkes process in (4).

Now we introduce our learning framework. Following the derivation for Vasicek process (Iacus, 2009), we will compute the marginal conditional density $p(\mathbf{x}_{v_{i+1}} | \mathbf{x}_{v_i})$ for each sample. In order to so, we first derive the close form solution of $\mathbf{x}(t_{i+1})$ given $\mathbf{x}(t_i)$ and conditional density:

Gaussian conditional density. Set $V(\mathbf{x}, t) = x \exp(t)$, apply the Generalized Ito's lemma in Theorem

2, and integrate from $[t_i, t_{i+1}]$, then we have

$$\mathbf{x}(t_{i+1}) = \mathbf{b} + (\mathbf{x}(t_i) - \mathbf{b}) \exp(-\Delta_i) + \beta \int_{t_i}^{t_{i+1}} \exp(-(t_{i+1} - s)) d\mathbf{w}(s) + \mathbf{h}(\mathbf{x}(t_i))(\mathbf{N}(t_{i+1}) - \mathbf{N}(t_i)) \exp(\Delta_i)$$

where $\Delta_i = t_{i+1} - t_i$. From the conditional law (Iacus, 2009), we can see $\mathbf{x}(t)$ has the Gaussian conditional density with mean and variance respectively:

$$\mathbb{E}[\mathbf{x}(t_{i+1})|\mathbf{x}(t_i)] = \mathbf{b} + (\mathbf{x}(t_i) - \mathbf{b}) \exp(-\Delta_i) + \mathbf{h}(\mathbf{x}(t_i))(\mathbf{N}(t_{i+1}) - \mathbf{N}(t_i)) \exp(\Delta_i) \quad (28)$$

$$\text{Var}[\mathbf{x}(t_{i+1})|\mathbf{x}(t_i)] = \beta^2(1 - \exp(-2\Delta_i))\mathbf{I} \quad (29)$$

Now we derive the marginal density of the above conditional Gaussian distribution at dimension v_{i+1} . Since only one event happened at time t_{i+1} to user v_{i+1} , the counting difference vector $\mathbf{N}(t_{i+1}) - \mathbf{N}(t_i)$ has value 1 at the v_{i+1} -th dimension and zero elsewhere. Hence the v_{i+1} -th dimension of $\mathbf{h}(\mathbf{x}(t_i))(\mathbf{N}(t_{i+1}) - \mathbf{N}(t_i))$ is $\mathbf{M}(v_i, v_{i+1})\mathbf{x}_{v_i}$, where $\mathbf{M}$ is the network adjacency matrix. Substitute it to (28) and get the variance of v_{i+1} -th dimension we have:

$$\begin{aligned} \mathbb{E}[\mathbf{x}_{v_{i+1}}|\mathbf{x}_{v_i}] &= \mathbf{b}_{v_i} + (\mathbf{x}_{v_i} - \mathbf{b}_{v_i}) \exp(-\Delta_i) + \mathbf{M}(v_i, v_{i+1})\mathbf{x}_{v_i} \\ \text{Var}[\mathbf{x}_{v_{i+1}}|\mathbf{x}_{v_i}] &= \beta^2(1 - \exp(-2\Delta_i)) \end{aligned}$$

We can compute $p(\mathbf{x}_{v_{i+1}}|\mathbf{x}_{v_i})$ from these two statistics.

Point process likelihood. Now we compute the probability that an event happens at v_{i+1} at t_{i+1} and no event happens between t_i and t_{i+1} as follows (Aalen et al., 2008):

$$P(\mathbf{N}_{v_{i+1}}(t_{i+1}) - \mathbf{N}_{v_{i+1}}(t_i) = 1) = \lambda_{v_i}(t_i) \prod_{v=1}^U \exp\left(-\int_{t_{i-1}}^{t_i} \lambda_v(t) dt\right)$$

The complete likelihood function for all samples $\mathcal{T} = \{t_i, \mathbf{x}_{v_i}, v_i\}$ is:

$$\ell(\mathcal{T}) = \underbrace{\prod_{i=1}^n p(\mathbf{x}_{v_{i+1}}|\mathbf{x}_{v_i})}_{\text{Gaussian density}} \underbrace{\prod_{v=1}^U \exp\left(-\int_0^T \lambda_v(t) dt\right)}_{\text{point process}}$$

For Hawkes process, we parametrize $\lambda(t)$ in (4). The parameters can be estimated by maximizing the likelihood, i.e., $\max_{\alpha \geq 0, \eta \geq 0, \beta \geq 0, \mathbf{b}} \ell(\mathcal{T})$.

We can see the likelihood function is nicely decomposed into two parts. The *Gaussian* part corresponds to Gaussian distribution using the property of Wiener process. The *point process* part is the likelihood for Hawkes processes. Moreover, since parameters $\{\eta, \alpha\}$ are linear in $\lambda(t)$ and parameters $\{\mathbf{b}, \beta\}$ are linear in the mean and standard deviation of Gaussian density, the overall objective is *concave*, and the global optimum can be found by many algorithms. In our experiments, we adapt the efficient algorithm developed for Hawkes process in previous work (Zhou et al., 2013) to update $\{\eta, \alpha\}$ and projected gradient descent to update $\{\mathbf{b}, \beta\}$.

9 Experiments

We evaluate the performance of our model on both synthetic and real-world datasets on the tasks :

1. **Efficient opinion control:** Evaluate how our model controls dynamics of both stable and unstable systems.
2. **Robust opinion control:** Evaluate how our model controls dynamics with inaccurate parameters.
3. **Control with node birth process:** Evaluate how our model controls dynamics with more and more user joining the network.
4. **Min-max control:** Evaluate the min-max control performance under different levels of disturbances in the differential game scenario.

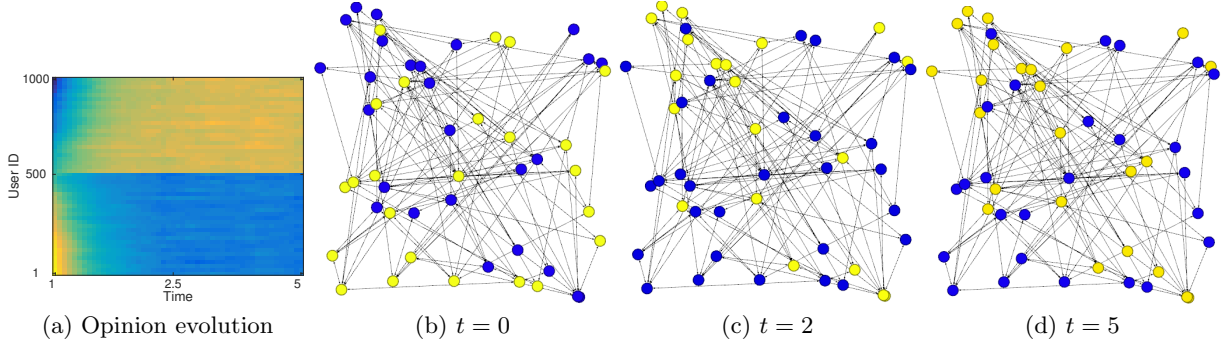


Figure 1: Controlled opinion of 1000 users with initial state uniformly sampled from $[-10, 10]$ and sorted, target state is polarized with -5 and 10 . (a) shows the opinion *value* per user over time. (b-d) are snapshots of the opinion *polarity* in the network with 50 sub-users. Yellow/blue means positive/negative.

9.1 Experiments on Synthetic Data

Experimental setup. We generate both stable and unstable systems. *Stable* means the opinion dynamics will converge to its expected value as time tends to infinity. First we simulate the Hawkes process with 1,000 users on observation window $[0, 50]$ using the thinning algorithm (Ogata, 1981). We equally divide the observation window into 500 segments $\{\tau_k\}_{k=0}^{500}$, with interval $\Delta t = 0.1$. Then we apply Euler forward approximation to the opinion dynamics in (2) with Wiener increments of ΔW_k and Hawkes increments ΔN_k . The diffusion part ΔW_k is sampled from the normal distribution $\mathcal{N}(0, \sqrt{\Delta t})$. And we compute ΔN_k by counting the number of events on $[\tau_k, \tau_{k+1})$.

For parameter settings, we set each user with the same initial opinion $\mathbf{x}_0 = -10$ and the target $\mathbf{a} = 1$. We set the noise level of Wiener process, $\beta = 0.2$. The network $\mathbf{M}$ is a fixed random matrix with sparsity of 0.001. Each entry of $\mathbf{M}$ is generated uniformly on $[0, 0.01]$. For Hawkes process parameters, we set $\boldsymbol{\eta} = \mathbf{0.01}$ and $\boldsymbol{\alpha}$ generated the same way as $\mathbf{M}$. For the stable system, we choose $\mathbf{M}, \boldsymbol{\alpha}, \mathbf{b}$ to satisfy the *stationary* conditions in (De et al., 2015).

Competitors. We compare our piecewise-constant control with the continuous and fully observed control, and *Stationary* (De et al., 2015), which will converge to the target opinion with our parameter settings for the *stable* system.

Efficient opinion control. Figure 2 shows the results for a 1000-user network. In (a), the controlled network achieves the target opinion *much faster* than that in (b). Moreover, we can see our method has less variance and much better control of the opinion jumps, *i.e.*, when a user tweets/retweets. (c) further demonstrates that our method has much lower path cost compared with the *Stationary* method. Since *Stationary* does not control the network, one’s opinion is easily influenced by others, hence large variance. However, for our controlled network, we are able to control the influence level of these events hence less variance. Finally, (d) demonstrates that the piecewise-constant control is a good approximation to the (best) continuous control. Moreover, for the unstable system that will not converge to the target opinion, (b) shows that the *Stationary* method will explode and (a) demonstrates that we are still able to control the network efficiently. Figure 1 visualizes the controlled network. In appendix we further show experiments with four different choices of initial state and target state. In summary, our control i) is comparable to the (best) continuous and fully observed control, ii) has less variance and fast convergence speed to the target, and iii) works efficiently for both *stable* and *unstable* system.

Robust opinion control. To control the dynamics, we will learn parameters from data. To investigate how the control performs with estimated parameters, we generate samples with 10 and 100 tweeting events per user for the 1000-user network and learn parameters. Then we compare the path cost between i) control with learned parameters, ii) control with true parameters, iii) *Stationary* with learned parameters. Figure 3 shows that even with *inaccurate* parameters, our control framework is still **robust** and controls the dynamics efficiently. However, the *Stationary* method will have high variance and could not steer the system towards the target state due to inaccurate parameters.

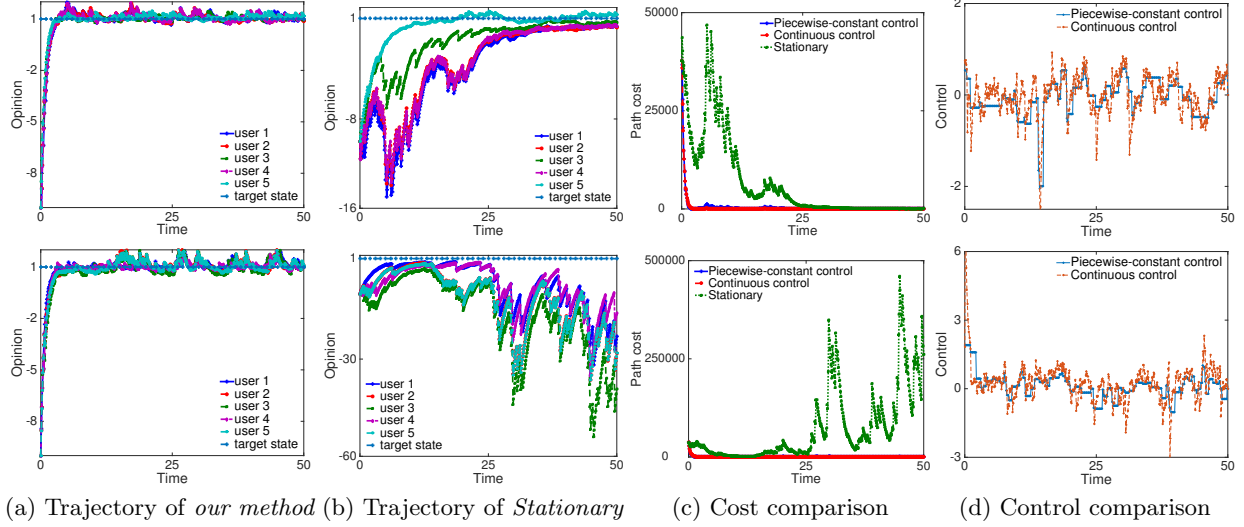


Figure 2: Experiments on a stable (1st row) and an unstable (2nd row) system. (a,b) show sample trajectories of 5 users. In (c) the cost for controlled dynamics is the sum of the variance from the target state and control cost, while that for *Stationary* is just the variance. The jumps in (a,b,c) correspond to the tweeting events. (d) compares our control with the best control for one sampled user.

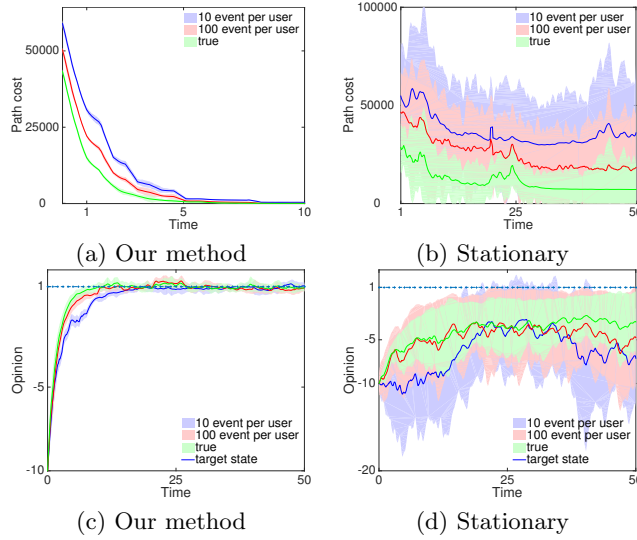


Figure 3: Comparison between model learned with different size of samples and true model. Path cost comparison in (a,b) is for all users and trajectory comparison in (c,d) is for one randomly sample user. The line is the mean and pale region is variance over ten runs.

9.2 Experiments on Real-world Data

We evaluate the performance of our control over node birth network on two real-world datasets. *Food*¹ contains top 500 users with most reviews and scores from 2001 to 2012, with 32,775 reviews. *Movie*² contains top 1000 users with most reviews from 1997 to 2012. The total number of reviews is 1,343,161. The raw review score ranges from 1 to 5 and we map it to $[-10, 10]$. These two datasets do not explicitly provide a social network and we build as follows. We assume two users follow each other if they have posted at least

¹<http://snap.stanford.edu/data/web-FineFoods.html>

²<http://snap.stanford.edu/data/web-Movies.html>

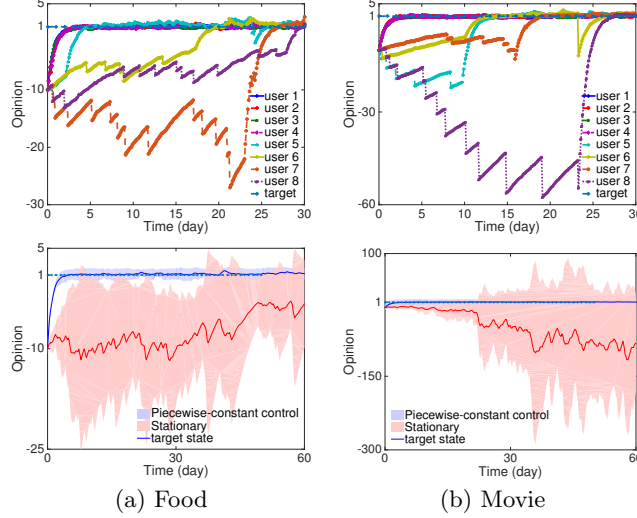


Figure 4: Experiments on real datasets. The first row shows opinion trajectories of 8 users on two datasets. User 1-4 are in the network and 5-8 join sequentially. The second row shows dynamics of one randomly sampled user over 10 runs.

5/10 reviews on the same type of food/movie. For *Food*, we use the data up to year 2010 to build the initial network with 447 users, and 53 new users joined the network since then. For *Movie*, we use the data before 2005 to generate the initial network with 623 users, and 377 new users joined the network since then. We first learn the model parameters and simulate the dynamics for two months and control over the simulated dynamics. For node birth process parameters, we learn the Poisson intensity γ by (Aalen et al., 2008). We learn the parameters for the dynamics by our parameter estimation scheme. We set initial state to be -10 and the target state to be 1 for each user.

Figure 4 shows our control model can steer the opinion of the user once he joins the network. Specifically, in (b), User 8 joins the network around day 25. Before this time, his opinion is not controlled and changes according to its own dynamics. He creates a link to user 6, since there is a immediate drop in user 6’s opinion. Again our control framework will steer their opinions to the target state quickly. Moreover, the averaged dynamics further shows that in the two datasets, as more and more nodes join the network, the two systems are **unstable** with large variance, but our framework still works efficiently.

9.3 Robustness of Min-max Control

We evaluate the robustness of the Min-max control on a synthetic 10-user network. The network $\mathbf{M}$ is a fixed random matrix with sparsity of 0.2. Each entry of $\mathbf{M}$ is generated uniformly on $[0, 0.4]$. For Hawkes process parameters, we set the baseline intensity $\boldsymbol{\eta} = \mathbf{0.1}$ and the mutual excitation coefficient $\boldsymbol{\alpha}$ generated the same way as $\mathbf{M}$. For the dynamics coefficients, we set $\beta = \theta = \theta_2 = 1$.

Figure 5 shows that the controlled trajectories for the no-opponent system are the same as these systems with different opponents. We can see from the first row of (a,b,c) that the stabilizing control $\mathbf{u}^*$ always adapts to its opponent $\mathbf{v}^*$ and changes its values accordingly. It works efficiently as if there were no opponents. Hence we conclude that the min-max control is robust with respect to different levels of disturbances.

10 Conclusion

We have proposed a novel opinion dynamics and control model which build a new bridge between social information diffusion and stochastic control theory. This allow us to leverage methods and results from stochastic control to address social information diffusion problems. There are many extension we can imagine along this direction. For instance, the jump term in the model can also be extended to marked point processes (Jacobsen, 2006). The dynamics of a linear SDE can also be extended to nonlinear cases (Theodorou, 2015)

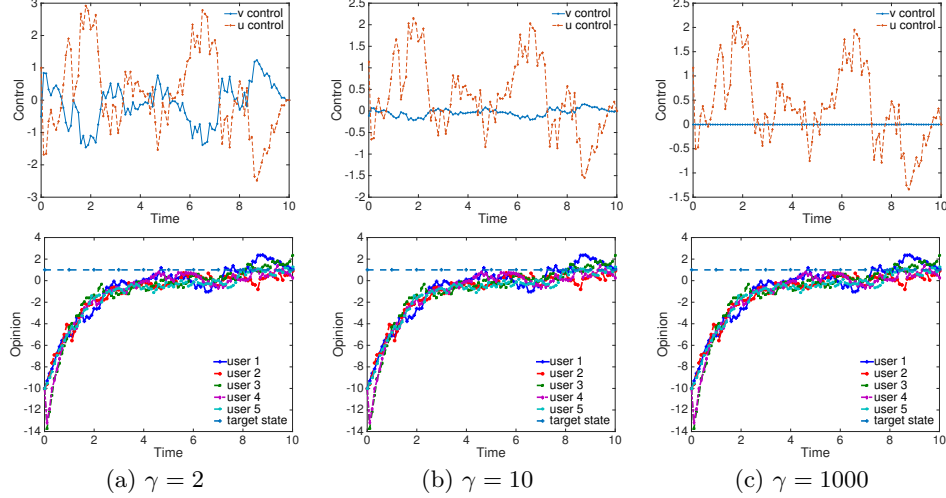


Figure 5: Experiments on a 10-user network with different values of γ . The first row corresponds to the optimal stabilizing control $\mathbf{u}^*$ and de-stabilizing control $\mathbf{v}^*$ of one randomly sampled user and the second row shows the opinion trajectories of 5 users. As γ becomes small, $\mathbf{v}^*$ becomes large, while the controlled dynamics remains same.

where one needs to generalize the path integral framework (Theodorou & Todorov, 2012) to the case of general point processes.

References

- Aalen, Odd, Borgan, Ornulf, and Gjessing, Hakon. *Survival and event history analysis: a process point of view*. Springer, 2008.
- Ades, Michel, Caines, Peter E, and Malhamé, Roland P. Stochastic optimal control under poisson-distributed observations. *Automatic Control, IEEE Transactions on*, 45(1):3–13, 2000.
- Clifford, Peter and Sudbury, Aidan. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.
- Cox, D.R. and Isham, V. *Point processes*, volume 12. Chapman & Hall/CRC, 1980.
- Cox, D.R. and Lewis, P.A.W. Multivariate point processes. *Selected Statistical Papers of Sir David Cox: Volume 1, Design of Investigations, Statistical Methods and Applications*, 1:159, 2006.
- Daley, D.J. and Vere-Jones, D. *An introduction to the theory of point processes: volume II: general theory and structure*, volume 2. Springer, 2007.
- Das, Abhimanyu, Gollapudi, Sreenivas, and Munagala, Kamesh. Modeling opinion dynamics in social networks. In *WSDM*, 2014.
- De, Abir, Valera, Isabel, Ganguly, Niloy, Bhattacharya, Sourangshu, and Rodriguez, Manuel Gomez. Modeling opinion dynamics in diffusion networks. *arXiv preprint arXiv:1506.05474*, 2015.
- DeGroot, Morris H. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- Dormand, John R and Prince, Peter J. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.
- Du, Nan, Wang, Yichen, He, Niao, and Song, Le. Time sensitive recommendation from recurrent user activities. In *NIPS*, 2015.
- Farajtabar, Mehrdad, Du, Nan, Gomez-Rodriguez, Manuel, Valera, Isabel, Zha, Hongyuan, and Song, Le. Shaping social activity by incentivizing users. In *NIPS*, 2014.

- Farajtabar, Mehrdad, Wang, Yichen, Gomez-Rodriguez, Manuel, Li, Shuang, Zha, Hongyuan, and Song, Le. Coevolve: A joint point process model for information diffusion and network co-evolution. In *NIPS*, 2015.
- Hanson, Floyd B. *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*, volume 13. Siam, 2007.
- Hegselmann, Rainer and Krause, Ulrich. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.
- Holme, Petter and Newman, Mark EJ. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74(5):056108, 2006.
- Hu, Yaozhong and Oksendal, Bernt. Partial information linear quadratic control for jump diffusions. *SIAM Journal on Control and Optimization*, 47(4):1744–1761, 2008.
- Iacus, Stefano M. *Simulation and inference for stochastic differential equations: with R examples*, volume 1. Springer Science & Business Media, 2009.
- Jacobsen, Martin. *Point process theory and applications*. Springer, 2006.
- Liniger, Thomas Josef. *Multivariate Hawkes Processes*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2009.
- Molin, Adam and Hirche, Sandra. Optimal event-triggered control under costly observations. In *Proceedings of the 19th international symposium on mathematical theory of networks and systems*, 2010.
- Morimoto, Jun and Atkeson, Christopher G. Minimax differential dynamic programming: An application to robust biped walking. In *NIPS*, 2003.
- Nazarzadeh, J, Razzaghi, M, and Nikraves, KY. Solution of the matrix riccati equation for the linear quadratic control problems. *Mathematical and Computer Modelling*, 27(7):51–55, 1998.
- Nguyen, Thang and Gajic, Zoran. Solving the matrix differential riccati equation: a lyapunov equation approach. *Automatic Control, IEEE Transactions on*, 55(1):191–194, 2010.
- Ogata, Yoshihiko. On lewis’ simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.
- Oksendal, Bernt Karsten and Sulem, Agnes. *Applied stochastic control of jump diffusions*, volume 498. Springer, 2005.
- Theodorou, Evangelos and Todorov, Emo. Stochastic optimal control for nonlinear markov jump diffusion processes. In *American Control Conference*, 2012.
- Theodorou, Evangelos A. Nonlinear stochastic control and information theoretic dualities: Connections, interdependencies and thermodynamic interpretations. *Entropy*, 17(5):3352–3375, 2015.
- Wang, Yichen and Pal, Aditya. Detecting emotions in social media: A constrained optimization approach. In *IJCAI*, 2015.
- Westman, JJ and Hanson, EB. The lqgp problem: A manufacturing application. In *American Control Conference, 1997. Proceedings of the 1997*, volume 1, pp. 566–570. IEEE, 1997.
- Yong, Jiongmin and Zhou, Xun Yu. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.
- Zhou, Ke, Zha, Hongyuan, and Song, Le. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTAT*, 2013.

A Derivations from HJB Equation to Three ODEs in (11), (12) and (13)

We restate the HJB equation as follows:

$$-\frac{\partial V}{\partial t} = \min_{\mathbf{u}} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \frac{\beta^2}{2} \text{tr}(V_{\mathbf{x}\mathbf{x}}(\mathbf{x}, t)) + V_{\mathbf{x}}(\mathbf{x}, t)^\top (\mathbf{b} - \mathbf{x}(t) + \theta \mathbf{u}(t)) + \sum_{j=1}^U \lambda_j(t) (V(\mathbf{x} + \mathbf{h}_j(\mathbf{x}), t) - V(\mathbf{x}, t)) \right\} \quad (30)$$

Substitute $\mathbf{u}^*$ in (10) to the HJB equation, we will compute the four terms in the right side of the HJB equation. Note that the minimization is reached when $\mathbf{u} = \mathbf{u}^*$. In the following computation, we will use the property that $\mathbf{v}_{11} = \mathbf{v}_{11}^\top$ and $\mathbf{a}^\top \mathbf{b} = \mathbf{b}^\top \mathbf{a}$ for any vector $\mathbf{a}$ and $\mathbf{b}$.

First, $\mathcal{L}(\mathbf{x}, \mathbf{u}^*, t)$ is:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}^*, t) = \frac{1}{2}(\mathbf{x}(t) - \mathbf{a})^\top (\mathbf{x}(t) - \mathbf{a}) + \frac{\rho}{2} \mathbf{u}^\top(t) \mathbf{u}(t) \quad (31)$$

$$\begin{aligned} &= \frac{1}{2} \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{a} + \frac{1}{2} \rho \mathbf{u}^{*\top} \mathbf{u}^* \\ &= \frac{1}{2} \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{a} + \frac{\theta^2}{2\rho} (\mathbf{v}_1 + \mathbf{v}_{11} \mathbf{x})^\top (\mathbf{v}_1 + \mathbf{v}_{11} \mathbf{x}) \\ &= \frac{1}{2} \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{a} + \frac{\theta^2}{2\rho} \mathbf{v}_1^\top \mathbf{v}_1 + \frac{\theta^2}{\rho} \mathbf{v}_1^\top \mathbf{v}_{11} \mathbf{x} + \frac{\theta^2}{2\rho} \mathbf{x}^\top \mathbf{v}_{11} \mathbf{v}_{11} \mathbf{x} \\ &= \underbrace{\frac{\theta^2}{2\rho} \mathbf{v}_1^\top \mathbf{v}_1}_{\text{scalar}} + \underbrace{\mathbf{x}^\top \left(\frac{\theta^2}{\rho} \mathbf{v}_{11} \mathbf{v}_1 - \mathbf{a} \right)}_{\text{linear}} + \underbrace{\frac{1}{2} \mathbf{x}^\top \left(\frac{\theta^2}{\rho} \mathbf{v}_{11} \mathbf{v}_{11} + \mathbf{I} \right) \mathbf{x}}_{\text{quadratic}} \end{aligned} \quad (32)$$

Note in line 2 of the expansion of $\mathcal{L}$, we dropped the constant term $\frac{1}{2} \mathbf{a}^\top \mathbf{a}$.

The second term is:

$$\text{tr}(V_{\mathbf{x}\mathbf{x}}(\mathbf{x}, t)) = \frac{\beta^2}{2} \text{tr}(\mathbf{v}_{11}) \quad (33)$$

The third term is:

$$\begin{aligned} V_{\mathbf{x}}^\top (\mathbf{b} - \mathbf{x} + \theta \mathbf{u}^*) &= (\mathbf{v}_1 + \mathbf{v}_{11} \mathbf{x})^\top (\mathbf{b} - \mathbf{x} - \theta \mathbf{u}^*) \\ &= (\mathbf{v}_1 + \mathbf{v}_{11} \mathbf{x})^\top (\mathbf{b} - \mathbf{x} - \frac{\theta^2}{\rho} (\mathbf{v}_1 + \mathbf{v}_{11} \mathbf{x})) \\ &= (\mathbf{v}_1^\top \mathbf{b} - \frac{\theta^2}{\rho} \mathbf{v}_1^\top \mathbf{v}_1) - (\mathbf{v}_1^\top \mathbf{x} + \frac{\theta^2}{\rho} \mathbf{v}_1^\top \mathbf{v}_{11} \mathbf{x} + \frac{\theta^2}{\rho} \mathbf{v}_1^\top \mathbf{v}_{11} \mathbf{x} - \mathbf{b}^\top \mathbf{v}_{11} \mathbf{x}) - \mathbf{x}^\top \mathbf{v}_{11}^\top \mathbf{x} - \frac{\theta^2}{\rho} \mathbf{x}^\top \mathbf{v}_{11}^\top \mathbf{v}_{11} \mathbf{x} \\ &= \underbrace{(\mathbf{v}_1^\top \mathbf{b} - \frac{\theta^2}{\rho} \mathbf{v}_1^\top \mathbf{v}_1)}_{\text{scalar}} - \underbrace{(\mathbf{v}_1^\top \mathbf{x} + \frac{2\theta^2}{\rho} \mathbf{v}_{11} \mathbf{v}_1 - \mathbf{v}_{11} \mathbf{b})}_{\text{linear}} - \underbrace{\frac{1}{2} \mathbf{x}^\top (2\mathbf{v}_{11} + \frac{2\theta^2}{\rho} \mathbf{v}_{11} \mathbf{v}_{11}) \mathbf{x}}_{\text{quadratic}} \end{aligned} \quad (34)$$

The fourth term is:

$$\begin{aligned} \sum_{j=1}^U \lambda_j(t) (V(\mathbf{x} + \mathbf{h}_j(\mathbf{x}), t) - V(\mathbf{x}, t)) &= \sum_{j=1}^U \lambda_j(t) (\mathbf{v}_1^\top \mathbf{A}_j \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{A}_j^\top \mathbf{v}_{11} \mathbf{A}_j \mathbf{x} + \frac{1}{2} \mathbf{x}^\top 2\mathbf{v}_{11} \mathbf{A}_j \mathbf{x}) \\ &= \underbrace{\mathbf{x}^\top \mathbf{\Lambda}_1^\top \mathbf{v}_1}_{\text{linear}} + \underbrace{\frac{1}{2} \mathbf{x}^\top \left(\sum_{j=1}^U \lambda_j \mathbf{A}_j^\top \mathbf{v}_{11} \mathbf{A}_j + 2\mathbf{v}_{11} \mathbf{\Lambda}_1 \right) \mathbf{x}}_{\text{quadratic}} \end{aligned} \quad (35)$$

where $\mathbf{\Lambda}_1(t) = \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j$.

Next, we compute the left side of HJB equation as:

$$-\frac{\partial V}{\partial t} = -v'_0(t) - \mathbf{x}^\top \mathbf{v}'_1(t) - \frac{1}{2} \mathbf{x}^\top \mathbf{v}'_{11}(t) \mathbf{x} \quad (36)$$

Then by comparing the coefficients for the scalar, linear and quadratic terms in both left side and right side of the HJB equation, we obtain three ODEs.

First, only consider all the coefficients quadratic in $\mathbf{x}$:

$$-\frac{d\mathbf{v}_{11}(t)}{dt} = \mathbf{I} + 2\mathbf{v}_{11}(t)(-1 + \mathbf{\Lambda}_1(t)) + \sum_{j=1}^U \lambda_j(t) \mathbf{A}_j^\top \mathbf{v}_{11}(t) \mathbf{A}_j - \frac{\theta^2}{\rho} \mathbf{v}_{11}(t) \mathbf{v}_{11}(t)$$

Second, consider the linear term:

$$-\frac{d\mathbf{v}_1(t)}{dt} = -\mathbf{a} + (-1 + \mathbf{\Lambda}_1^\top(t) - \frac{\theta^2}{\rho} \mathbf{v}_{11}(t)) \mathbf{v}_1(t) + \mathbf{v}_{11}(t) \mathbf{b} \quad (37)$$

Third, consider the scalar term:

$$-\frac{dv_0(t)}{dt} = \mathbf{b}^\top \mathbf{v}_1(t) + \frac{\beta^2}{2} \text{tr}(\mathbf{v}_{11}(t)) - \frac{\theta^2}{2\rho} \mathbf{v}_1^\top(t) \mathbf{v}_1(t) \quad (38)$$

Finally, we compute the terminal condition for the three ODEs by $V(\mathbf{x}(T), T) = \phi(\mathbf{x}(T), T)$:

$$V(X(T), T) = v_0(T) + \mathbf{x}(T)^\top \mathbf{v}_1(T) + \frac{1}{2} \mathbf{x}(T)^\top \mathbf{v}_{11}(T) \mathbf{x}(T) \quad (39)$$

$$\phi(\mathbf{x}(T), T) = -\mathbf{x}(T)^\top \mathbf{a} + \frac{1}{2} \mathbf{x}(T)^\top \mathbf{x}(T) \quad (40)$$

Hence $v_0(T) = 0$, $\mathbf{v}_1(T) = -\mathbf{a}$ and $\mathbf{v}_{11} = \mathbf{I}$. Note here we drop the constant term $\frac{1}{2} \mathbf{a}^\top \mathbf{a}$ in terminal cost ϕ .

B Additional Experiments on Synthetic Datasets

We conduct control over four networks with different initial and target states. Figure 6 shows our framework works efficiently.

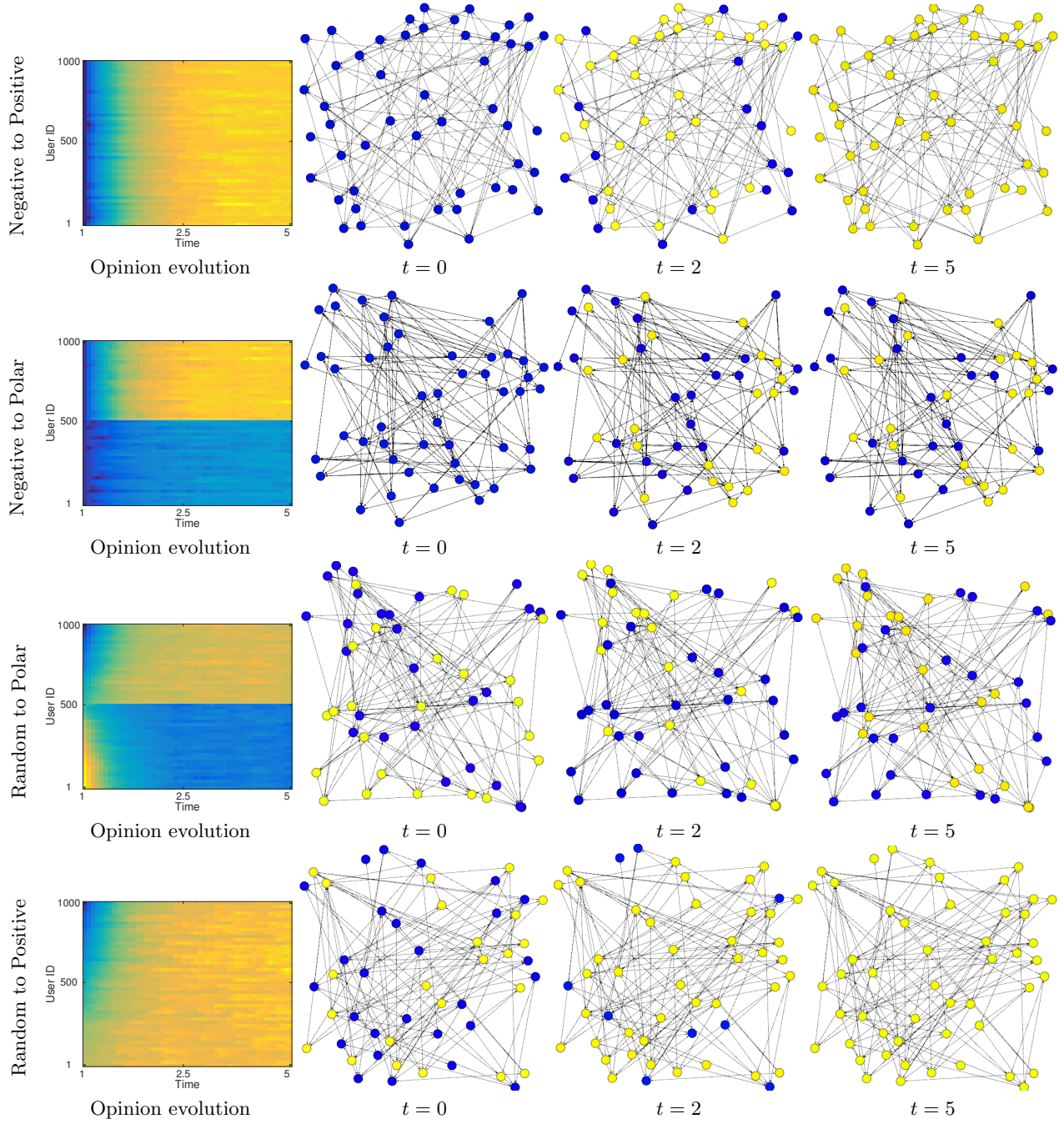


Figure 6: Controlled opinion of four networks with 1,000 users. The first column is the description of opinion change. The second column shows the opinion *value* per user over time. The three right columns show three snapshots of the opinion *polarity* in the network with 50 sub-users at different times. Yellow means positive and blue means negative polarity. Since the controlled trajectory converges fast, we use time range of $[0, 5]$. Parameters are same as these for Figure 2 except for different initial and target state: Set index I to denote user 1-500 and II to denote the rest. 1st row: $\mathbf{x}_0 = -10, \mathbf{a} = 10$. 2nd row: $\mathbf{x}_0 = -10, \mathbf{a}(\text{I}) = -5, \mathbf{a}(\text{II}) = 10$. 3rd row: $\mathbf{x}_0$ sampled uniformly from $[-10, 10]$ and sorted in decreasing order, $\mathbf{a}(\text{I}) = -10, \mathbf{a}(\text{II}) = 5$. 4th row: $\mathbf{x}_0$ is same as (c), $\mathbf{a} = 10$.

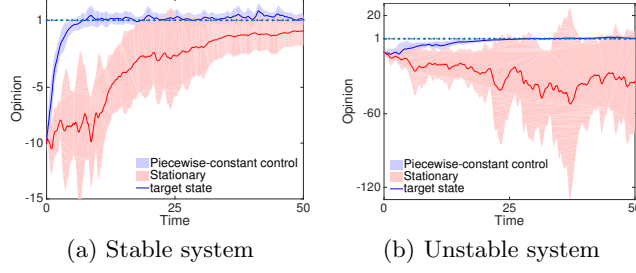


Figure 7: Dynamics of one randomly sampled user in the network with 1000 users, for both stable and unstable system. The red/blue line is the mean and pale region is the variance.

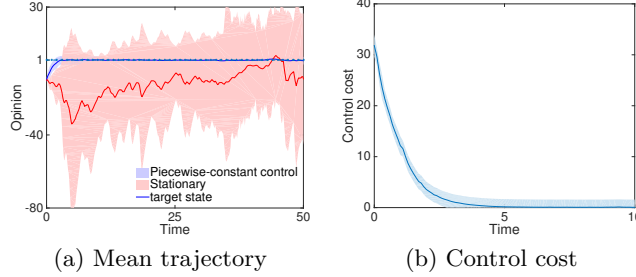


Figure 8: Experiments with control over a 1000-user network with link creation process. (b) is the control cost over all users.

Efficient control. Following the experiment settings in Figure 2, In Figure 7, we also average the experiments over ten runs and report the mean and variance of one randomly picked user’s opinion. It further shows that our method works efficiently for both stable and unstable system.

Control with time-changing network. We use the same parameter setting as that in fixed network case and set the initial network to be the one used for the stable system. Then we use the Poisson process intensity of $\gamma = 0.1$ to model the link creation process. Note that as more and more links are created, the system will change from stable to unstable. In order to compare with our method, for the *Stationary* method, we update the network structure whenever it changes. Figure 8 shows that our method can control this case well with quite low control cost.