

A JUMP STOCHASTIC DIFFERENTIAL EQUATION APPROACH FOR INFLUENCE PREDICTION ON HETEROGENOUS NETWORKS*

YAOHUA ZANG[†], GANG BAO[‡], XIAOJING YE[§], HONGYUAN ZHA[¶], AND HAOMIN ZHOU^{||}

Abstract. We propose a novel problem formulation of continuous-time information propagation on heterogeneous networks based on jump stochastic differential equations (JSDE). The structure of the network and activation rates between nodes are naturally taken into account in the JSDE. This new formulation allows for efficient and stable algorithms for a variety of challenging information propagation problems, including estimations of individual activation probability and influence level, by solving the JSDE numerically. In particular, we develop an efficient numerical algorithm for solving the JSDE by incorporating variance reduction; and furthermore, we provide theoretical bounds for its sample complexity. Numerical experiments on a variety of propagation networks show that the proposed method is more accurate and efficient compared with the state-of-the-art methods, and more importantly it can be applied to solve other critical information propagation problems to which existing methods cannot be applied.

Keywords. Propagation network; stochastic differential equation with jump; heterogeneous network; influence prediction; complexity.

AMS subject classifications. 65Y10; 94A20; 90B18.

1. Introduction

Information propagation on networks is a prevalent phenomenon in the real world [3, 19, 20]. Examples of information propagation include news spreading on social media [8, 9, 30], viral marketing [13, 14, 31], computer malware spread, and epidemic of contagious diseases [1, 17, 20, 22]. More specifically, for instance, a piece of information (such as news) can be retweeted by users (nodes) on the Twitter social network formed by their followee-follower relationships. In this case, we call a node activated, become active, or infected, if the user participates to tweet, and the followers of this user get activated if they retweet his/her tweet later. By this, the activation process gradually progresses and the tweet spreads out. Such information propagation behaves very similarly as epidemic spread where a type of virus can infect an individual (human, animal, or plant) and spreads to others upon their close contact.

To begin with, we describe the basic information propagation model on a network, called diffusion network in [12]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a given network, i.e., directed graph, where $\mathcal{V} = \{1, \dots, n\}$ is the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. We denote $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. In addition, we denote $\mathcal{A} := \{\alpha_{ij} > 0 : (i, j) \in \mathcal{E}\}$, where $\alpha_{ij} > 0$ is the activation/infection rate of i on j . More precisely, once the node i becomes active at time t_i , the time that i needs to infect its healthy neighbor j , denoted by $t_{ij} := t_j - t_i > 0$, follows the $\text{Exp}(\alpha_{ij})$ distribution. Here by $t \sim \text{Exp}(\alpha)$, we mean that the probability density of t is $f(t) = \alpha e^{-\alpha t}$ for $t > 0$. Now suppose i (called the source node) is active at time 0, then it will start to activate its neighbors in $\mathcal{V}_i^- := \{j : (i, j) \in \mathcal{E}\}$

*Received: July 03, 2019; Accepted (in revised form): July 10, 2020. Communicated by Qiang Du.

[†]School of Mathematics, Zhejiang University, 38 Zheda Road, Hangzhou, China (yhchuang@zju.edu.cn).

[‡]School of Mathematics, Zhejiang University, 38 Zheda Road, Hangzhou, China (baog@zju.edu.cn).

[§]Department of Mathematics and Statistics, Georgia State University, 25 Park Place, Atlanta, GA, USA (xye@gsu.edu).

[¶]School of Data Science, Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, 2001 Longxiang Road, Shenzhen, China (zhahy@cuhk.edu.cn).

^{||}School of Mathematics, Georgia Institute of Technology, North Avenue, Atlanta, GA, USA (hmzhou@math.gatech.edu).

simultaneously and independently, each at the corresponding rate given in $\{\alpha_{ij} \in \mathcal{A} : j \in \mathcal{V}_i^-\}$. An activated neighbor node j will then start to activate its healthy neighbors in \mathcal{V}_j^- at the corresponding rates, and so on. Thus the information originated from the node i can propagate to other nodes on the network. In a slight generalization to include recovery scenario, an active/infected node i may also recover at some rate $\gamma_i > 0$, become inactive/healthy and prune to activation/infection again.

Given the aforementioned propagation model, we are interested in the fundamental problem of influence prediction in this paper. More specifically, we want to compute the influence, defined as the expected number of active nodes on the network, at any time $t > 0$ given that the propagation started from a known set of source nodes. Moreover, we are also interested in the probability that a specific node i is active/infected at time t . However, these seemingly simple problems turn out to be very challenging computationally: the heterogeneous structure of the given network \mathcal{G} and the variations in activation rates in \mathcal{A} (and recovery rates $\mathcal{R} := \{\gamma_i > 0 : i \in \mathcal{V}\}$ if applicable) significantly complicate the computation. For example, an analytic solution of influence prediction is shown to require computation of a Markov chain whose state space is of size $O(2^n)$ [12, 27], which quickly becomes computationally intractable as n increases.

The influence prediction problem can be significantly more complicated even with slight modification to the basic propagation model above. For example, if the activation time t_{ij} follows a non-exponential distribution, such as Weibull, Rayleigh, or power-law, the propagation is no longer Markov and an analytic solution is not available. Another example is that the activation processes are not independent: the rate of j being infected is not simply the sum of α_{ij} 's over its infected parents in $\mathcal{V}_j^+ := \{i : (i, j) \in \mathcal{E}\}$, but some nonlinear function of them. In this case, even direct simulation of the propagation becomes computationally prohibitive.

In this paper, we propose a novel approach to address the aforementioned computational problem of influence prediction. Our approach is based on a reformulation of the information propagation on heterogeneous network into a system of jump stochastic differential equations (JSDEs) [2, 21]. In contrast to existing methods, our formulation provides a JSDE prospective of the interdependent activation processes between nodes on heterogeneous networks, based on which we develop an efficient numerical algorithm with variance reduction for influence prediction and individual node activation probability estimation with and without recovery. Our method is instantiated using the basic propagation model with exponentially distributed activation times for demonstration purpose, whereas enhancement to further extend our approach is also provided. More importantly, we show that our approach can be applied to a variety of other critical information propagation problems where some or all existing methods may fail to work.

The rest of this paper is organized as follows. First we review the literature on information propagation on networks that are related to this work in Section 2. In Section 3, we provide the details of the proposed JSDE reformulation. Then we develop a numerical algorithm based on the new formulation to solve influence prediction problem in Section 4. We present applications of the proposed method to more general propagations in Section 5. We demonstrate the performance of the proposed algorithm, with comparison to several state-of-the-arts methods, on a variety of networks in Section 6. Section 7 concludes the paper.

2. Related work

The basic information propagation model with constant activation (infection) rates is equivalent to the classical susceptible-infected (SI) model (or a variation, called susceptible-infected-susceptible, or SIS, model where an infected node can recover and

become prone to infection again) which has been extensively studied in the past decades for *statistically homogenous* networks where individual nodes are indistinguishable. A recent comprehensive survey can be found in [20]. In contrast, existing work on SI/SIS for fixed and heterogeneous networks considered in this paper is very limited, mainly due to the significant complications and computational challenges due to the prescribed diversities of nodes and their interactions. In this case, a solution to exactly describe the process requires a state space of size $O(2^n)$, where n is the size of the network, and hence is computationally intractable in practice [12, 27]. As an alternative, Monte-Carlo (MC) simulation can be used to sample the activation time on each edge and then form a sample cascade of events. This is repeated for many times to obtain an averaged result that approximates the influence. However, this MC approach requires extensive amount of samples to achieve satisfactory approximation accuracy, and hence is only used to generate ground truth for reference during comparison in the literature [8, 12].

One of the major approaches to *approximately* quantify influence or infection probability for the basic propagation model with constant infection/recovery rates is based on mean-field theory. In [26, 27], mean-field approximation is applied to reduce the $O(2^n)$ linear system describing the Markov SIS process to an n -intertwined model, which is a system of n nonlinear ordinary differential equations. This method adopts a first-order moment closure that ignores dependencies between infection states of neighbor nodes, and hence its solution gives an upper bound of infection probability for each node. Mean-field approximation is then also applied to more complicated cases including multi-layer network [23], weighted network [32], and hypergraph [1]. The mean-field approach is also applied to the competing bi-virus model in [15]. In [22], an additional “alert” state of nodes is considered, such that individuals with infected neighbors may enter an alert state and become less prone to infection. In [4], a second-order mean-field approximation is employed which improves the estimate of epidemic threshold over first-order mean-field at the cost of significantly more computation complexity. In [29], the authors propose an accuracy criterion of mean-field approximation using the covariance between infection states of adjacent nodes. However, it is computationally intractable to estimate this error due to its exponentially large size. In [5], a discrete Fokker-Planck equation based on aggregated activation states is proposed which yields fast and accurate computation of influence without the presence of recovery. Discussions on limitations of mean-field are also provided in [10]. In [28], the authors showed that the basic propagation model with exponentially-distributed infection time can be unrealistic in real-world applications and the Markov property that mean-field approaches are based on may not hold.

In recent years, there is a significant increase of interests in information propagation in network and data science due to its prevalent applications in social networking and cyber security. Most literature in this field focuses on *discrete-time* information propagations where infections can occur only at discrete time points. In contrast, continuous-time information propagation studied in the literature closely mimics the SI/SIS model but is also much more challenging computationally as mentioned above. In [12], an analytic solution is derived based on the observation that infection time is the length of the stochastic shortest path from the source nodes to the node. This method establishes Markov chains for each node and can estimate individual infection probability, however, the complexity still grows at order $O(2^n)$ for general networks and hence is not scalable. In [7, 8], the authors propose a novel sampling technique to estimate the coverage function in information propagation, and developed efficient algorithms to approximate influence even for large networks. In [24], the authors derived

bounds of influence and characterized phase transition using the spectral radii of the Laplace Hazard matrix. Learnability of influence function, which is the core of influence prediction in these methods, is considered in [18].

3. JSDE formulation of information propagation

In this section, we propose a novel formulation for the information propagation on network using jump stochastic differential equations. This new formulation has an exact and concise mathematical interpretation of the complex random propagation process. Moreover, we provide an efficient numerical algorithm to solve the influence prediction problem based on this formulation.

For ease of presentation, we first focus on the basic propagation model where the activation times follow independent exponential distributions as described in Section 1. That is, the time t_{ij} for a just-activated node i to activate/infect its inactive/healthy neighbor j follows the $\text{Exp}(\alpha_{ij})$ distribution. In the presence of recovery, a just-activated node i can also recover and become inactive at rate γ_i , i.e., the time needed for recovery follows the $\text{Exp}(\gamma_i)$ distribution. We assume all these activation/infection and recovery times are independent.

To represent the propagation process as a system of JSDEs, we first denote the stochastic process $X_i(t)$ as the time-evolving activation state of the node i at time t . Namely, $X_i(t)=1$ if node i is active/infected at time t , and $X_i(t)=0$ otherwise. Therefore, each $X_i(t)$ is a right-continuous function of time t with left limit. We denote $X_i(t^-):=\lim_{\tau\rightarrow t^-} X_i(\tau)$ the left limit of X at t . For each edge $(i,j)\in\mathcal{E}$ associated with activation rate α_{ij} , we introduce an auxiliary temporal point process $N_{ij}(t)\sim\text{PP}(\alpha_{ij})$. Here, by $N(t)\sim\text{PP}(\alpha)$ we meant that $N(t)$ is the time-homogeneous Poisson process with intensity α , namely, $\lim_{\delta\rightarrow 0^+}(1/\delta)\cdot\mathbb{E}[N(t+\delta)-N(t)]=\alpha$ for all t . In other words, $N(t)$ can be thought of as a counting process whose value is 0 at time $t=0$ and increases (jumps) by 1 at each time τ_1,τ_2,\dots , where $\{\tau_{k+1}-\tau_k:k\geq 0\}$ are i.i.d. $\text{Exp}(\alpha)$ random variables ($\tau_0=0$ by convention). Hence $N(t)=k$ if $\tau_k\leq t<\tau_{k+1}$. In the presence of recovery, we also introduce $R_i(t)\sim\text{PP}(\gamma_i)$ for each $i\in\mathcal{V}$. Note that $\{N_{ij}(t),R_i(t):i\in\mathcal{V},(i,j)\in\mathcal{E}\}$ is a finite set of Poisson processes, and hence two or more of them jumping at the same time has probability 0.

Now the key observation is that we can think $N_{ij}(t)$ as of i ‘‘sending an activation signal’’ to j at times τ_1,τ_2,\dots . Therefore, $dN_{ij}(t)=1$ if $t=\tau_k$ for some k or otherwise $dN_{ij}(t)=0$. We note that a node j becomes activated by i at t successfully if and only if $X_i(t^-)=1$, $X_j(t^-)=0$, and $dN_{ij}(t)=1$. By this we have $dX_i(t)=1$, and $X_i(t)$ jumps from 0 to 1. Considering that there may be multiple parent nodes in \mathcal{V}_j^+ sending activation signals to j simultaneously, we can write the activation process of j as

$$dX_j(t)=(1-X_j(t^-))\sum_{i\in\mathcal{V}_j^+}X_i(t^-)dN_{ij}(t) \tag{3.1}$$

for every $j=1,\dots,n$. In the presence of recovery, we know an active node j becomes deactivated/recovered if and only if $X_j(t^-)=1$ and $dR_j(t)=1$. Therefore, we can add this recovery term to (3.1) and obtain

$$dX_j(t)=\left[(1-X_j(t^-))\sum_{i\in\mathcal{V}_j^+}X_i(t^-)dN_{ij}(t)\right]-X_j(t^-)dR_j(t) \tag{3.2}$$

for $j=1,\dots,n$. Compared to (3.1), the additional term $-X_j(t^-)dR_j(t)$ in (3.2) indicates that $X_j(t)$ can jump from 1 to 0 ($dX_j(t)=-1$) if j recovers at time t .

To rewrite (3.2) in concise matrix form, we first introduce the following vectors (unless otherwise noted, all vectors are column vectors):

$$\mathbf{X}(t)=(X_1(t),\dots,X_n(t))^T\in\mathbb{R}^n \tag{3.3}$$

$$\mathbf{N}_{\cdot j}(t) = (N_{k_1j}(t), N_{k_2j}(t), \dots, N_{k_{|\mathcal{V}_j^+|}j}(t))^\top \in \mathbb{R}^{|\mathcal{V}_j^+|} \tag{3.4}$$

$$\mathbf{N}(t) = (\mathbf{N}_{\cdot 1}(t)^\top, \dots, \mathbf{N}_{\cdot n}(t)^\top)^\top \in \mathbb{R}^m \tag{3.5}$$

$$\mathbf{R}(t) = (R_1(t), \dots, R_n(t))^\top \in \mathbb{R}^n \tag{3.6}$$

$$\mathbf{J}(t) = (\mathbf{R}(t)^\top, \mathbf{N}(t)^\top)^\top \in \mathbb{R}^{n+m} \tag{3.7}$$

where $k_1 < k_2 < \dots$ are the parent nodes of j in \mathcal{V}_j^+ . Furthermore, $\forall \mathbf{X} = (X_1, \dots, X_n)^\top \in \mathbb{R}^n$, we define matrix functions $\mathbf{c}_0(\mathbf{X}), \mathbf{c}_1(\mathbf{X}), \mathbf{c}(\mathbf{X})$ as

$$\mathbf{c}_0(\mathbf{X}) = \text{diag}(X_1, \dots, X_n) \in \mathbb{R}^{n \times n} \tag{3.8}$$

$$\mathbf{c}_1(\mathbf{X}) = \text{diag}(\mathbf{b}_1(\mathbf{X})^\top, \dots, \mathbf{b}_n(\mathbf{X})^\top) \in \mathbb{R}^{n \times m} \tag{3.9}$$

$$\mathbf{c}(\mathbf{X}) = [-\mathbf{c}_0(\mathbf{X}), \mathbf{c}_1(\mathbf{X})] \in \mathbb{R}^{n \times (n+m)} \tag{3.10}$$

where $\mathbf{b}_j(\mathbf{X}) = ((1 - X_j)X_{k_1}, \dots, (1 - X_j)X_{k_{|\mathcal{V}_j^+|}})^\top \in \mathbb{R}^{|\mathcal{V}_j^+|}$. Note that $\mathbf{c}_0(\mathbf{X})$ in (3.8) is a diagonal matrix, and $\mathbf{c}_1(\mathbf{X})$ in (3.9) is a block-diagonal matrix with row vectors $\mathbf{b}_j(\mathbf{X})^\top$ as the (j, j) -block.

Using the vector and matrix notations above, we can rewrite (3.2) concisely as follows:

$$d\mathbf{X}(t) = \mathbf{c}(\mathbf{X}(t^-))d\mathbf{J}(t) . \tag{3.11}$$

The initial $\mathbf{X}(0) = (X_1(0), \dots, X_n(0))^\top$ is determined such that $X_i(0) = 1$ if i is a source node and 0 otherwise. Note that, without recovery, the system (3.11) reduces to

$$d\mathbf{X}(t) = \mathbf{c}_1(\mathbf{X}(t^-))d\mathbf{N}(t), \tag{3.12}$$

which is equivalent to (3.1).

We remark that the system of n coupled JSDEs, (3.11), or equivalently (3.2), represents the basic propagation model exactly. Since the stochastic process $X_i(t)$ is binary-valued, we know that the probability of a node i being active at time t is $x_i(t) := \mathbb{E}[X_i(t)]$. Moreover, the influence, defined by the expected number of active nodes at time t , is $\mu(t) := \mathbb{E}[\mathbf{1}^\top \mathbf{X}(t)] = \mathbb{E}[\sum_i X_i(t)] = \sum_i x_i(t)$. Therefore, the system (3.11) and its solution $\mathbf{X}(t)$ play the central role of our algorithmic development for influence prediction below.

4. Algorithm and complexity

As showed above, the basic information propagation process can be formulated as the system of JSDEs (3.2). To obtain individual activation probability $x_i(t)$ and influence $\mu(t)$, we need to solve the JSDE system (3.2) numerically to estimate $\mathbb{E}[X_i(t)]$ and $\mathbb{E}[\mathbf{1}^\top \mathbf{X}(t)]$, respectively. In the literature, numerical approximations to the solution of a JSDE can be generally categorized in two types: strong approximation and weak approximation. Strong approximation is used to estimate the solution $\mathbf{X}(t)$ pathwise. Weak approximation, on the other hand, is used to compute the expectations of (functions of) $\mathbf{X}(t)$, such as $\mathbb{E}[X_i(t)]$ and $\mathbb{E}[\mathbf{1}^\top \mathbf{X}(t)]$. Therefore, we use weak approximations in this paper as it suffices for our influence prediction problems.

In what follows, we first introduce the (weak) Euler and Taylor schemes with standard time discretization for solving (3.11) in Section 4.1. Then we employ a variance reduction technique and present our algorithm in Section 4.2. In Section 4.3, we establish the relation between approximation error and sampling complexity of the proposed algorithm.

4.1. Euler and Taylor schemes. We consider the solution of the system (3.11) over time interval $[0, T]$ for some prescribed time horizon T . For ease of presentation, we partition $[0, T]$ into $K(h) := T/h$ equal segments using discretization points $t_k = kh$ for $k = 0, 1, \dots, K(h)$, where without loss of generality we assume the step size $h \in (0, 1)$. Now we want to compute $\mathbf{X}^h(t)$ which approximate $\mathbf{X}(t)$ at all t_k for $k = 0, 1, \dots, K(h)$. For notation simplicity, we often denote $\mathbf{X}_k^h := \mathbf{X}^h(t_k)$ in the derivation below.

DEFINITION 4.1 (Weak order). An approximation \mathbf{X}^h is said to converge to \mathbf{X} at weak order $\beta > 0$ if for any $g \in C_P^{2\beta+2}(\mathbb{R}^{n+m}; \mathbb{R})$, there exists a constant $C_{g,T} > 0$ independent of h such that

$$|\mathbb{E}[g(\mathbf{X}^h(T))] - \mathbb{E}[g(\mathbf{X}(T))]| \leq C_{g,T} h^\beta. \tag{4.1}$$

for all $h > 0$ sufficiently small. Here $C_P^\beta(\mathbb{R}^n; \mathbb{R})$ denotes the space of β times continuously differentiable functions $\mathbb{R}^n \rightarrow \mathbb{R}$ with partial derivatives with polynomial growth.

Note that the weak order β of convergence for a discretization scheme has important practical consequences for simulation efficiency.

To generate \mathbf{X}^h that approximates the solution \mathbf{X} in (3.11), a basic algorithm is the Euler scheme:

$$\mathbf{X}_{k+1}^h = \mathbf{X}_k^h + \mathbf{c}(t_k, \mathbf{X}_k^h) \Delta \mathbf{J}_k, \tag{4.2}$$

where we also included t in function \mathbf{c} so that the method can generalize to cases where \mathbf{c} is time-dependent. In (4.2), the increment $\Delta \mathbf{J}_k = \mathbf{J}(t_{k+1}) - \mathbf{J}(t_k)$ is a vector of independent Poisson random numbers. More precisely, $\Delta \mathbf{J}_k = (\Delta \mathbf{R}_k^\top, \Delta \mathbf{N}_k^\top)^\top \in \mathbb{R}^{n+m}$ where the components of $\Delta \mathbf{R}_k$ and $\Delta \mathbf{N}_k$ are generated from Poisson distributions with $\{\gamma_i h\}$ and $\{\alpha_{ij} h\}$ as parameters, respectively. The Euler scheme (4.2) is known to be convergent at weak order $\beta = 1$ [21].

By adding more terms in the Wagner-Platen expansion [21], we can obtain numerical schemes with a higher order of convergence. For example, the Taylor scheme with convergence of weak order $\beta = 2$ is given by

$$\begin{cases} \mathbf{X}_{k+.5}^h = \mathbf{X}_k^h + \mathbf{c}_k \Delta \mathbf{J}_k \\ \mathbf{X}_{k+1}^h = \mathbf{X}_k^h + \mathbf{c}_k \Delta \mathbf{J}_k + (1/2) \cdot (\mathbf{c}_{k+.5} - \mathbf{c}_k) \Delta \mathbf{J}_k (\Delta \mathbf{J}_k - 1) \end{cases} \tag{4.3}$$

where $\mathbf{c}_k := \mathbf{c}(t_k, \mathbf{X}_k^h)$, $\mathbf{c}_{k+.5} := \mathbf{c}(t_k, \mathbf{X}_{k+.5}^h)$, and the subtraction and multiplication in $\Delta \mathbf{J}_k (\Delta \mathbf{J}_k - 1)$ are executed component-wise to obtain a vector in \mathbb{R}^{m+n} .

Higher-order Taylor schemes can also be obtained by adding more Wagner-Platen terms to (4.3). We refer interested readers to [21]. In this paper, we only use the Euler scheme (4.2) with $\beta = 1$, and sometimes the Taylor scheme (4.3) with $\beta = 2$, since they are accurate and cost-effective for our influence prediction problem.

It is also worth noting that, for mark-independent pure JSDEs such as (3.11), jump-adapted time discretization can directly simulate jump times for non-uniform time discretization. More precisely, in jump-adapted discretization, we first simulate a trajectory $0 = \tau_0 < \tau_1 < \dots \leq T$ of the Poisson process $\mathbf{J}(t)$, and apply it to the JSDE (3.11) so that the next sample $\hat{\mathbf{X}}(\tau_{k+1})$ can be computed directly given $\hat{\mathbf{X}}(\tau_k)$. For $t \in [\tau_k, \tau_{k+1})$ the approximation $\hat{\mathbf{X}}(t)$ remains as constant $\hat{\mathbf{X}}(\tau_k)$. One advantage of such jump-adapted Euler scheme is that it does not generate discretization error, and it is particularly efficient for low-intensity jump processes. In this work, we will incorporate the jump-adapted discretization into the regular uniform discretization of the Euler and Taylor schemes in our algorithm.

4.2. Variance reduction. The numerical solution \mathbf{X}^h obtained by the Euler (4.2) or Taylor (4.3) schemes is one sample approximation of the stochastic process \mathbf{X} defined in (3.11). To estimate $\mathbb{E}[g(\mathbf{X}^h(T))]$ in the applications of influence prediction and individual activation probability, we can employ (4.2) or (4.3) for L times to obtain $\{\mathbf{X}^{h,l} : 1 \leq l \leq L\}$, and use their sample mean to approximate $\mathbb{E}[g(\mathbf{X}^h(T))]$. More specifically, for the prescribed time horizon T , the sample mean $u_T(h, L)$ is defined by

$$u_T(h, L) = (1/L) \cdot \sum_{l=1}^L g(\mathbf{X}^{h,l}(T)) \tag{4.4}$$

which depends on both the discretization step size h and the number of sample trajectories L . This sample mean $u_T(h, L)$ is our approximation to $\mathbb{E}[g(\mathbf{X}^h(T))]$.

As we will show later, the mean square error of the approximation $u_T(h, L)$ depends on its variance $(1/L) \cdot \text{var}(g(\mathbf{X}^{h,l}(T)))$. To reduce this approximation error in practice, we employ a variance reduction technique introduced in [16] in our sampling algorithm. More specifically, in each sampled propagation (also called sampling trajectory), we generate a pair of antithetical samples Z^+ and Z^- from each Poisson random variable of mean $\gamma_i T$ and $\alpha_{ij} T$ for $i \in \mathcal{V}$ and $(i, j) \in \mathcal{E}$, respectively. Then we sample Z^+ and Z^- points independently and uniformly on $[0, T]$ respectively. Thus each set of points forms a trajectory of the Poisson process, which can be used in the Euler scheme (4.2) or the Taylor scheme (4.3) to obtain a sample of solution $\mathbf{X}_K^h = \mathbf{X}^h(t_K) = \mathbf{X}^h(T)$ to (3.11). The antithetical property of Z^\pm reduces the variance of $\mathbf{X}^h(t)$ in practice. The algorithm is summarized in Algorithm 1. For ease of presentation, we assume that the sample size L is even, and the step size h is chosen such that T/h is an integer.

Algorithm 1 Sample approximation $u_T(h, L)$ of influence $\mu_{g,T} = \mathbb{E}[g(\mathbf{X}(T))]$

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\{\alpha_{ij}, \gamma_i : i \in \mathcal{V}, (i, j) \in \mathcal{E}\}$, and T . Set h, L .

Set $K = L/h$ and $t_k := [kh, (k+1)h)$ for $k = 0, \dots, K-1$. Set $s = 0$.

for $l = 1, 2, \dots, L/2$ **do**

for each $i \in \mathcal{V}$ **do**

 Set F to the cumulative distribution of $\text{Poisson}(\gamma_i T)$;

 Draw $U \sim \text{Uniform}(0, 1)$, and set $Z^+ = F^{-1}(1 - U)$ and $Z^- = F^{-1}(U)$;

 Sample Z^+ and Z^- points, $\tau_{\gamma_i}^l := \{\tau_z^l : 1 \leq z \leq Z^+\}$ and $\tau_{\gamma_i}^{(L/2)+l} := \{\tau_z^{L/2+l} : 1 \leq z \leq Z^-\}$ respectively, each on $[0, T]$ uniformly;

 Set $(\Delta \mathbf{R}_k^\ell)_i = |\tau_{\gamma_i}^\ell \cap t_k|$ for $k = 0, \dots, K-1$ and $\ell = l, (L/2) + l$;

end for

for each $(i, j) \in \mathcal{E}$ **do**

 Set F to the cumulative distribution of $\text{Poisson}(\alpha_{ij} T)$;

 Draw $U \sim \text{Uniform}(0, 1)$, and set $Z^+ = F^{-1}(1 - U)$ and $Z^- = F^{-1}(U)$;

 Sample Z^+ and Z^- points, $\tau_{\alpha_{ij}}^l := \{\tau_z^l : 1 \leq z \leq Z^+\}$ and $\tau_{\alpha_{ij}}^{(L/2)+l} := \{\tau_z^{L/2+l} : 1 \leq z \leq Z^-\}$ respectively, each on $[0, T]$ uniformly;

 Set $(\Delta \mathbf{N}_k^\ell)_{ij} = |\tau_{\alpha_{ij}}^\ell \cap t_k|$ for $k = 0, \dots, K-1$ and $\ell = l, (L/2) + l$;

end for

 Solve (4.2) or (4.3) for $\mathbf{X}_K^{h,\ell}$ using $\{\Delta \mathbf{R}_k^\ell, \Delta \mathbf{N}_k^\ell : k\}$ for $\ell = l, (L/2) + l$;

$s \leftarrow s + g(\mathbf{X}_K^{h,l}) + g(\mathbf{X}_K^{h,(L/2)+l})$;

end for

Output: $u_T(h, L) = s/L$.

4.3. Sample complexity analysis. To establish the relation between approximation accuracy and cost, we provide a comprehensive analysis of computation and sampling complexity for our JSDE-based influence prediction method. In particular, for any fixed time horizon $T > 0$ and influence evaluation function g , we will derive upper bound of the root mean squared error (RMSE), denoted by $e_T(h, L)$, of the sample approximation $u_T(h, L)$ to $\mu_{g,T} := \mathbb{E}[g(\mathbf{X}(T))]$. The RMSE $e_T(h, T)$ is defined by

$$e_T(h, L) = \left\{ \mathbb{E} \left[|u_T(h, L) - \mu_{g,T}|^2 \right] \right\}^{1/2}. \tag{4.5}$$

Without loss of generalization, we again assume that $h \in (0, 1)$ and T/h is an integer. In addition, our complexity analysis requires the following conditions.

ASSUMPTION 4.1. *The stochastic process $\mathbf{X}(t)$ and the influence evaluation function g satisfy:*

- (1) *The function g satisfies polynomial growth. Namely, $\exists C > 0$ and positive integer $s > 0$ such that $g(\mathbf{x}) \leq C(1 + \|\mathbf{x}\|^s)$ for all $\mathbf{x} \in \mathbb{R}^n$.*
- (2) *The influence $g(\mathbf{X}(t))$ has bounded second moment, i.e., $\mathbb{E}[g^2(\mathbf{X}(T))] < \infty$.*

Now we are ready to present the first result that links the RMSE to the step size h and the number of sampling trajectories L in the approximation $u_T(h, L)$.

THEOREM 4.1. *Let $u_T(h, L)$ be the sample approximation to $\mu_{g,T}$ generated by Algorithm 1 with a numerical JSDE scheme (4.2) or (4.3) of weak order $\beta > 0$. Suppose Assumption 4.1 holds for g and $\mathbf{X}(t)$. Then there exists a constant $\sigma_{g,T}^2$ dependent on g and T but not on h and β , such that*

$$e_T(h, L) \leq \left(\frac{\sigma_{g,T}^2}{L} + C_{g,T}^2 h^{2\beta} \right)^{1/2}. \tag{4.6}$$

Proof. As g has polynomial growth, so does g^2 . Therefore we know there exists $C > 0$, independent of β , such that $\text{var}(g(\mathbf{X}^h(T)))$, the variance of $g(\mathbf{X}^h(T))$, has the following bound:

$$\text{var}(g(\mathbf{X}^h(T))) \leq \mathbb{E}[g^2(\mathbf{X}^h(T))] \leq \mathbb{E}[g^2(\mathbf{X}(T)) + Ch^\beta] \leq \mathbb{E}[g^2(\mathbf{X}(T))] + C =: \sigma_{g,T}^2, \tag{4.7}$$

where the second inequality is due to $|\mathbb{E}[g^2(\mathbf{X}^h(T))] - \mathbb{E}[g^2(\mathbf{X}(T))]| \leq Ch^\beta$ for some C independent of h by Theorem 12.3.4 in [21], and the last inequality is due to $h \in (0, 1)$ and $\beta > 0$. Furthermore, the RMSE $e_T(h, L)$ of $u_T(h, L)$ defined in (4.5) satisfies

$$\begin{aligned} e_T^2(h, L) &= \mathbb{E}[|u_T(h, L) - \mu_{g,T}|^2] \\ &= \mathbb{E}[|u_T(h, L) - \mathbb{E}[g(\mathbf{X}^h(T))] + \mathbb{E}[g(\mathbf{X}^h(T))] - \mu_{g,T}|^2] \\ &= \mathbb{E}[|u_T(h, L) - \mathbb{E}[g(\mathbf{X}^h(T))]|^2 + |\mathbb{E}[g(\mathbf{X}^h(T))] - \mu_{g,T}|^2] \\ &\leq \mathbb{E}[|u_T(h, L) - \mathbb{E}[g(\mathbf{X}^h(T))]|^2] + C_{g,T}^2 h^{2\beta} \\ &= (1/L) \cdot \text{var}(g(\mathbf{X}^h(T))) + C_{g,T}^2 h^{2\beta} \\ &\leq (1/L) \cdot \sigma_{g,T}^2 + C_{g,T}^2 h^{2\beta}, \end{aligned}$$

where we used the fact $\mathbb{E}[u_{h,L}(T)] = \mathbb{E}[g(\mathbf{X}^h(T))]$ in the third equality, (4.1) in the first inequality, and the fact that every $\mathbf{X}^{h,l}$ has the same distribution as \mathbf{X}^h in the fifth equality, and (4.7) in the last inequality. Taking square root on both sides yields (4.6). □

COROLLARY 4.1. *Suppose the conditions in Theorem 4.1 hold. For any $\epsilon > 0$, the RMSE satisfies $e_T(h, L) \leq \epsilon$ if the step size h and the number of trajectories L are set to*

$$h = T\delta_{g,T,\beta} D_{g,T,\beta}^{-1/\beta} \epsilon^{1/\beta} \quad \text{and} \quad L = \delta_{g,T,\beta} D_{g,T,\beta}^2 \epsilon^{-2}, \tag{4.8}$$

where the constants $\delta_{g,T,\beta}$ and $D_{g,T,\beta}$ only depend on g, T, β as follows,

$$\delta_{g,T,\beta} = \left(\frac{\sigma_{g,T}^2}{2\beta C_{g,T}^2 T^{2\beta}} \right)^{1/(2\beta)} \quad \text{and} \quad D_{g,T,\beta} = \left(\frac{\sigma_{g,T}^2}{\delta_{g,T,\beta}} + C_{g,T}^2 T^{2\beta} \delta_{g,T,\beta}^{2\beta} \right)^{1/2}. \tag{4.9}$$

In particular, there is $e_T(h, L) \leq \epsilon$ if the total sampling complexity is

$$\left(\sum_{i,j} \alpha_{ij} + \sum_i \gamma_i \right) T \delta_{g,T,\beta} D_{g,T,\beta}^2 \epsilon^{-2} = O(\epsilon^{-2}) \tag{4.10}$$

on expectation and the computation complexity is

$$O((m+n)(\epsilon/D_{g,T,\beta})^{-(2\beta+1)/\beta}) = O(\epsilon^{-(2\beta+1)/\beta}). \tag{4.11}$$

Proof. As the total computation complexity is linear in T/h and L , we denote the cost $B = LT/h$. For any B , the minimum of the bound of $e_T(h, L)$ in (4.6) can be obtained by solving $\min_{h,L} \{(\sigma_{g,T}^2/L) + C_{g,T}^2 h^{2\beta}\}$ subject to $B = LT/h$, which yields solution

$$h = T\delta_{g,T,\beta} B^{-1/(2\beta+1)} \quad \text{and} \quad L = \delta_{g,T,\beta} B^{2\beta/(2\beta+1)}, \tag{4.12}$$

where $\delta_{g,T,\beta}$ is defined in (4.9). In this case, the bound given in (4.6) can be written as $e_T(h, L) \leq D_{g,T,\beta} B^{-\beta/(2\beta+1)}$. In order to have $e_T(h, L) \leq \epsilon$, it suffices to have $B = (\epsilon/D_{g,T,\beta})^{-(2\beta+1)/\beta}$, which together with (4.12) yields (4.8).

In Algorithm 1, each trajectory of (4.2) or (4.3) needs to have $(\sum_{i,j} \alpha_{ij} + \sum_i \gamma_i)T$ sampled points on expectation. Hence the total sampling complexity of L trajectories is $(\sum_{i,j} \alpha_{ij} + \sum_i \gamma_i)TL$, which together with L (4.8) yields (4.10). In addition, each step of (4.2) or (4.3) is $O(m+n)$, and hence the total computational complexity is $O((m+n)LT/h)$. Substituting h and L by the values in (4.8) yields (4.11). This completes the proof. \square

5. JSDE for more general propagation problems

The basic propagation model with constant activation (and recovery) rates discussed above is widely used in a variety of applications including news and disease spread etc. However, in many real-world applications, the propagations are often time and/or state dependent and the basic model with constant activations rates is not accurate. In these cases, the vast existing methods relying on the constant rates in basic propagation model are not suitable. On the other hand, our method based on JSDE can be readily modified to handle these cases by making the coefficients and jump intensity time and state dependent. In this section, we depict such generalization of our method to two scenarios. For conciseness of the present paper, we will report more in-depth analysis and numerical experiments of these cases in a forthcoming work.

5.1. Time varying activation rates. In some real-world applications, the impact of i on j may be diminishing along time, which mimics the phenomenon that older news/message makes less impact to a user's action. In this case, the activation rate $\alpha_{ij}(t)$ is time varying, for example, can be modeled as decaying such that $t_{ij} =$

$t_j - t_i \sim \text{Weibull}(\alpha_{ij}, \beta_{ij})$ for some $\beta_{ij} > 0$ instead of $\text{Exp}(\alpha_{ij})$. Here by $t \sim \text{Weibull}(\alpha, \beta)$ we mean that the probability density of t is $f(t) = \beta\alpha^\beta t^{\beta-1} e^{-(\alpha t)^\beta}$ for $t > 0$. This yields a time-varying activation rate $\alpha_{ij}(t) = \beta_{ij}\alpha_{ij}^{\beta_{ij}} t^{\beta_{ij}-1}$ of i on j where t is the time since i got activated. Note that with $\beta_{ij} = 1$ there is $t_{ij} \sim \text{Exp}(\alpha_{ij})$ and $\alpha_{ij}(t) \equiv \alpha_{ij}$, which reduces to the basic propagation model. Similar modifications can be made to the recovery rate so that $\gamma_i(t)$ is time varying.

Time-varying activation rates cause significant computational challenge for existing methods, since the propagation is no longer Markov. Moreover, Monte Carlo simulation becomes difficult and computationally demanding due to the dependency of propagation on its entire history. However, our approach can easily address this issue by incorporating an additional variable into the JSDE system. More specifically, we introduce an auxiliary variable $U_j(t)$ for every node j , and establish a system of JSDEs of $\{U_j(t), X_j(t) : j \in \mathcal{V}, t \in [0, T]\}$. For ease of presentation, we consider time-varying activation rates but still assume constant recovery rates, since further generalization is trivial.

We first observe that the key to incorporating the time dependence of activation rates is to record the time elapsed since the last activation of every node j . This time is denoted by $U_j(t)$, and the coupled JSDE system of $U_j(t)$ and $X_j(t)$ is given below:

$$\begin{cases} dX_j(t) = \left[(1 - X_j(t^-)) \sum_{i \in \mathcal{V}_j^+} X_i(t^-) dN_{ij}(t, U_i(t^-)) \right] - X_j(t^-) dR_j(t) \\ dU_j(t) = X_j(t^-) dt - U_j(t^-) dR_j(t) \end{cases} \quad (5.1)$$

where $U_j(t)$ is the time since last activation of j , and $U_j(t) = 0$ if j is currently inactive at time t . To see this, we first observe that $X_j(t)$ is binary-valued at 0 and 1 to indicate the activation state of j at time t . Therefore, the instantaneous rate of $U_j(t)$ is given by $X_j(t^-)$ and hence $U_j(t)$ accumulates at rate 1 when j is active, whereas the value of $U_j(t)$ drops to 0 every time j recovers, i.e., $X_j(t^-) dR_j(t) = 1$. In (5.1), the intensity of the Poisson process $N_{ij}(t, U_i(t^-))$ depends on U_i , the time since the last activation of the parent node i . For example, if the activation time follows the Weibull distribution with α_{ij} and β_{ij} as parameters, then we can obtain the intensity of $N_{ij}(t, U_i(t))$ as $\beta_{ij}\alpha_{ij}^{\beta_{ij}} (U_i(t))^{\beta_{ij}-1}$. Derivations for other types of distributions are similar.

5.2. State-dependent propagation. In some applications, the rate $\alpha_j(t)$ for a node j to get activated at time t is not simply the sum of α_{ij} over its active parents. Instead, α_j can be a nonlinear function of these α_{ij} . For example, the rate for j to get activated is $\alpha_j(t) = \min\{a_j, \sum_i \alpha_{ij} X_i(t^-)\}$ as a nonlinear function of activation states of its parents. Namely, $a_j > 0$ is a personal threshold such that the rate α_j is throttled and more active parents will not further increase the rate α_j . In this case, we can write the processes $N_{ij}(t)$ to be state dependent, i.e., $N_{ij}(t) = N_{ij}(t, \mathbf{X}(t^-))$, with intensity $\min\{a_j, \sum_i \alpha_{ij} X_i(t^-)\}$. The JSDE solver can be employed in the same way.

6. Numerical experiments

To demonstrate the effectiveness of the proposed method, we conduct extensive numerical experiments of influence prediction using various networks. More specifically, we test the proposed method on influence prediction problems on artificially generated networks for which ground truth influence can be obtained by large number of naive Monte Carlo simulations. For comparison purpose, we also implemented two state-of-the-art influence estimation methods and plot their results to show the improved efficiency and flexibility of the proposed method.

6.1. Experiment setup. To generate networks in our experiments, we used the CONTEST package [25] which is freely available online to the public. The CONTEST package can generate many types of networks given network size and specific parameters. We conducted our numerical tests by using different types of networks, and found that the influence prediction results are very similar. Therefore, for sake of conciseness, we only show the results using three types of networks: Erdős-Rényi network, small-world network, and scale-free (preferential attachment) network in this paper, as they are typical networks and are widely used in real-world applications. Unless otherwise noted, the source set of nodes are randomly selected, and fix it for all compared methods in each test.

The ground truth influence of each test is obtained by simulating a large number of propagations using Monte Carlo (MC) method and taking the sample mean. More specifically, we generate 10,000 propagations for a given network and source set selection, and obtain the empirical probability that node i is activate at time t , denoted $x_i(t)$. The total influence is obtained by $\mu(t) = \sum_i x_i(t)$. Therefore, $\mu(t)$ and $x_i(t)$ correspond to influence evaluation functions $g(\mathbf{x}) = \mathbf{1}^\top \mathbf{x}$ and $g_i(\mathbf{x}) = x_i$ respectively. In other words, $\mathbb{E}[g(\mathbf{X}(t))] = \sum_{i=1}^n \mathbb{E}[X_i(t)]$ is the expected number of active nodes at time t , and $\mathbb{E}[g_i(\mathbf{X}(t))] = \mathbb{E}[X_i(t)]$ is the probability that node i is active at time t . The estimated values are denoted by $\hat{\mu}(t)$ and $\hat{x}(t)$ respectively.

For comparison purpose, we also implemented two state-of-the-art methods to estimate influence on heterogeneous networks: the mean-field approximation (labeled as **Mean Field**) with the first-order moment closure [27] and a sampling-based approximation method for continuous-time influence estimation (labeled as **ConTinEst**) [8]. Mean Field approximates $x_i(t)$ by solving the following deterministic system of n coupled nonlinear differential equations numerically using e.g., 4th order Runge-Kutta method:

$$x'_j(t) = \left[(1 - x_j(t)) \sum_{i \in \mathcal{V}_j^+} \alpha_{ij} x_i(t) \right] - \gamma_j x_j(t), \quad j = 1, \dots, n. \tag{6.1}$$

The computational cost of Mean Field is very low since no stochasticity is involved. It is also worth noting that (6.1) can be deduced by taking expectation on both sides of the JSDE system (3.2), (incorrectly) ignoring all correlations between terms in multiplications, and using the facts that $\mathbb{E}[dN_{ij}(t)] = \alpha_{ij} dt$, $\mathbb{E}[dR_j(t)] = \gamma_j(t) dt$ and $\mathbb{E}[X_i(t^-)] = x_i(t)$. More specifically, the product $\mathbb{E}[X_i(t^-)X_j(t^-)]$ being replaced by $\mathbb{E}[X_i(t^-)]\mathbb{E}[X_j(t^-)]$ is due to the first-order moment closure as mentioned in [27] from a different point of view, and hence the result of Mean Field is overestimating the true influence. Unfortunately, the error caused by such moment-closure cannot be estimated in general and hence it is unclear how to further improve the accuracy of Mean Field. ConTinEst, on the other hand, is a recently developed approximation method based on effective samplings. It employs the Least-Label-List technique presented in [6]. ConTinEst can be applied to information propagation with infection time distribution other than exponential. However, as the influence is estimated based on coverage function, ConTinEst cannot estimate probability of an individual’s activation state, nor the influence of propagations with recovery.

6.2. Experiments on synthetic data. In the first experiment, we test on the influence prediction problem using the basic propagation model (i.e., activation rates) without recovery scenario. We first generate three networks: ER, small-world, and scale-free networks, each with $n = 200$ nodes. For Erdős-Rényi network, we randomly generate $m = \lceil n \log(n)/2 \rceil$ (where $\lceil a \rceil$ denotes the integer closest to $a \in \mathbb{R}$) edges and form

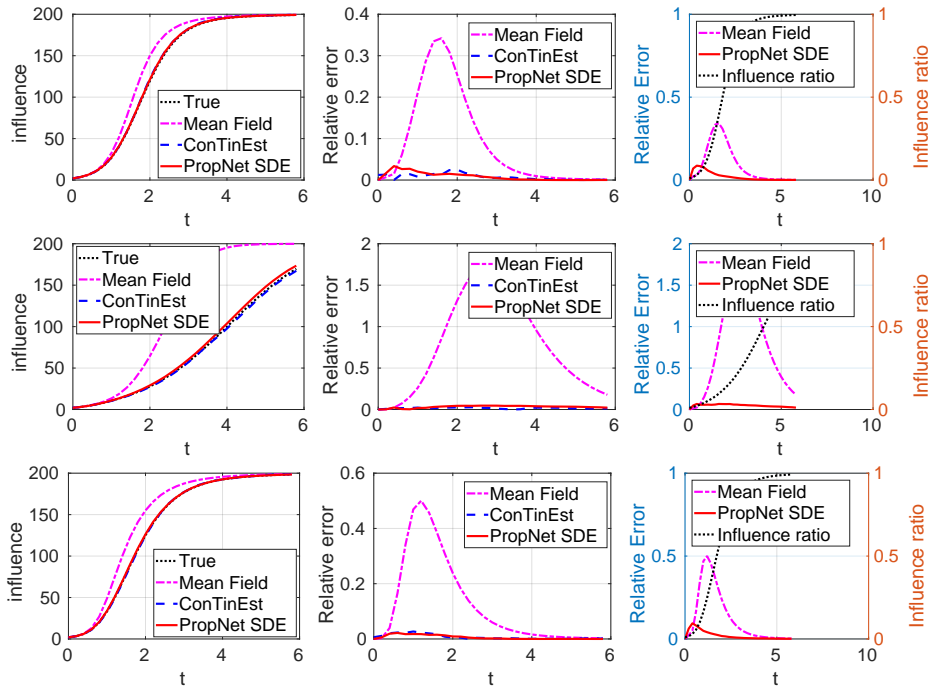


FIG. 6.1. Influence prediction by compared methods on Erdős-Rényi network (top row), small-world network (middle row), and scale-free network (bottom row), all of size $n=200$, using the basic propagation model *without* recovery. Left column: True total influence $\mu(t) = \sum_i x_i(t)$ and influences $\hat{\mu}(t) = \sum_i \hat{x}_i(t)$ obtained by Mean Field, ConTinEst, and PropNet SDE. Middle column: relative error in influence $|\hat{\mu}(t) - \mu(t)|/\mu(t)$. Right column: relative error in individual activation probability $\sum_i |\hat{x}_i(t) - x_i(t)|/\sum_i x_i(t)$ (influence ratio $\mu(t)/n$ is plotted in black dotted line for reference).

a directed graph. For small-world network, we start with a ring graph, and for each node we create a new link with probability 0.2 and connect it to another node on the network randomly chosen with uniform distribution. For scale-free (preferential attachment) network, we add nodes one by one to the existing network, where each new node has 2 links to the existing nodes, and the probability of linking to an existing node is proportional to the degree of that node. For each network, we choose two nodes at random to form the source set. Then we simulate 10,000 propagations using Monte Carlo method and compute the empirical mean of influence as the true influence (labeled as **True**). For ConTinEst method, we sample 1,000 trajectories and simulate 15 random labels in each trajectory. For the proposed JSDE method, i.e., Algorithm 1, we fix the time step size $h=0.01$ and sample $L=1,000$ trajectories. For the basic propagation model without recovery, the result of the compared methods on these three networks are shown in Figure 6.1.

From Figure 6.1, we can see that the proposed method generates highly accurate predictions as the estimated influence $\hat{\mu}(t)$ and individual activation probability $\hat{x}_i(t)$ have smallest relative error to the true $\mu(t)$ and $x_i(t)$ respectively (shown in the middle and right columns in Figure 6.1) in every test network. This is in sharp contrast to Mean Field which yields much larger error than the proposed PropNet SDE. ConTinEst also accurately estimated $\mu(t)$, however, it cannot estimate the individual activation probability $x_i(t)$ as the proposed PropNet SDE.

The second experiment is set similarly as the first one, but we incorporate recovery

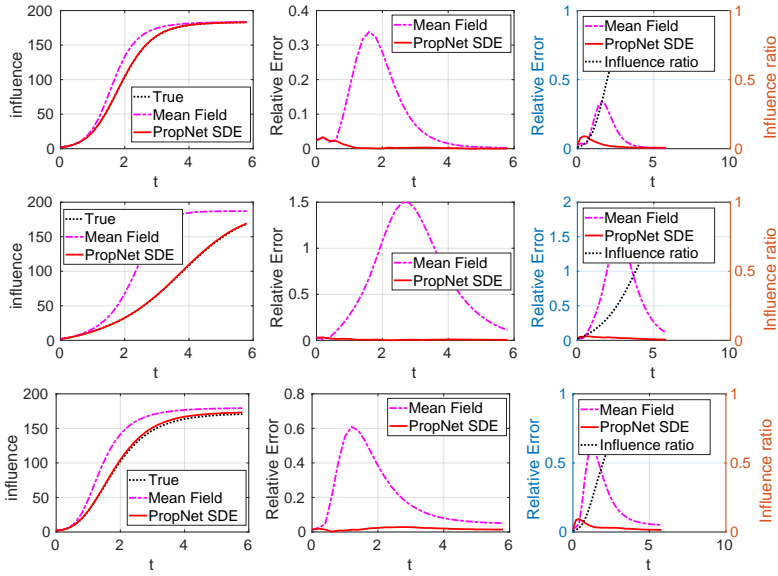


FIG. 6.2. Influence prediction by compared methods on Erdős-Rényi network (top row), small-world network (middle row), and scale-free network (bottom row), all of size $n=200$, using the basic propagation model with recovery. Left column: True total influence $\mu(t) = \sum_i x_i(t)$ and influences $\hat{\mu}(t) = \sum_i \hat{x}_i(t)$ obtained by Mean Field and PropNet SDE. Middle column: relative error in influence $|\hat{\mu}(t) - \mu(t)|/\mu(t)$. Right column: relative error in individual activation probability $|\sum_i \hat{x}_i(t) - x_i(t)|/\sum_i x_i(t)$ (influence ratio $\mu(t)/n$ is plotted in black dotted line for reference).

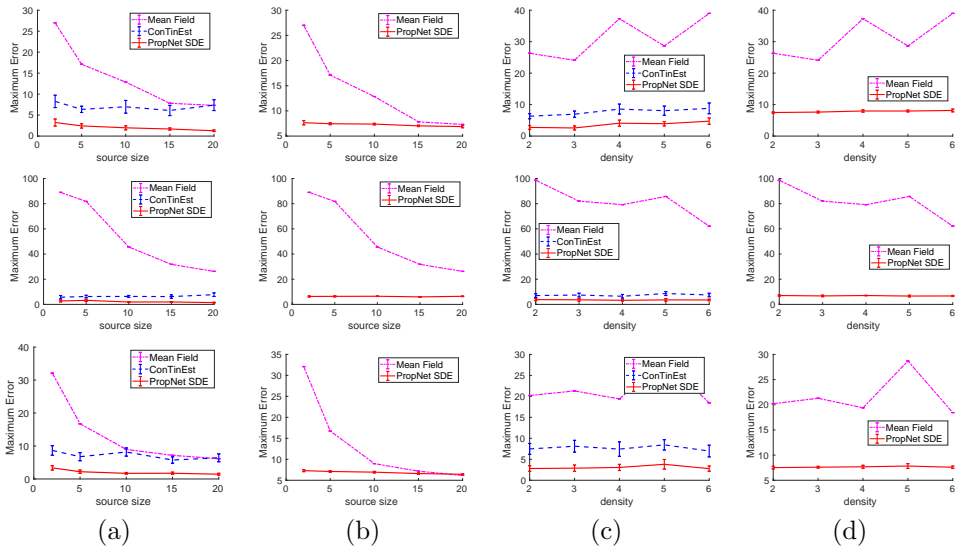


FIG. 6.3. Robustness test of compared methods on Erdős-Rényi network (top row), small-world network (middle row), and scale-free network (bottom row). Column (a): the maximum absolute error $\max_{0 \leq t \leq T} |\hat{\mu}(t) - \mu(t)|$ versus different sizes of source set using the basic propagation model without recovery; Column (b): the maximum absolute error versus different sizes of source set using the basic propagation model with recovery; Column (c): the maximum absolute error versus different density levels using the basic propagation model without recovery; Column (d): the maximum absolute error versus different density levels using the basic propagation model with recovery.

scenario such that an active node i can recover at a constant rate γ_i . In this test, the recovery rates for each node are chosen uniformly from $(0, 0.4)$. Since ConTinEst cannot handle the case with recoveries, we only compare Mean Field and PropNet SDE in this test. The results are shown in Figure 6.2. We again observe that PropNet SDE has significant improvement over Mean Field in terms of accuracy, since the relative errors to $\mu(t)$ and $x_i(t)$ using PropNet SDE are much smaller than that using Mean Field.

In the third experiment, we evaluate the robustness of the compared methods when the size of source set and network density vary. For the first part of this experiment, we use the same three networks in the first and second experiments, and set five different sizes of source set: $n_0 = 2, 5, 10, 15, 20$. For each of the three networks and each of the five different n_0 , we randomly select n_0 nodes as the source set, and apply the compared methods to estimate influence of the basic propagation without recovery on the network and source set combination. Such test is repeated for 20 times for each of the network and source set size combination, and the maximum absolute error, defined by $\max_{0 \leq t \leq T} |\hat{\mu}(t) - \mu(t)|$, for all methods are shown in the first column of Figure 6.3(a). In Figure 6.3(a), the standard deviation of ConTinEst and PropNet SDE are also shown at each source size (Mean Field is deterministic and hence no variance). From Figure 6.3(a), we can see that PropNet SDE produces the lowest absolute error among the compared methods, which indicates that it outperforms the others in terms of accuracy. Figure 6.3(b) shows the result on the basic propagation model with recovery scenario, where we again observe that PropNet SDE is much more accurate than Mean Field, and ConTinEst is not capable to handle this situation.

In the second part of the third experiment, we generate the same three types of networks with different density levels (i.e., average node degree). For Erdős-Rényi network, the number of edges is $m = \lceil n \log(\kappa n) / 2 \rceil$ for $\kappa = 2, 3, 4, 5, 6$ respectively. For small-world network, the starting network is a regular graph where each node is connected to its respectively $\kappa = 2, 3, 4, 5, 6$ nearest neighbors and the probability of creating a short cut of each node is set to 0.2. For scale-free network, the number of links that each newly added node has is set to $\kappa = 2, 3, 4, 5, 6$ respectively. For each network type, we generate a network using each of those density parameters κ and apply ConTinEst and PropNet SDE for 20 times to obtain their means and standard deviations of maximum absolute error. The results of the error versus the density parameter κ by the compared methods are given in Figure 6.3 (c) for the case without recovery and (d) for the case with recovery, respectively. From these plots, we can see that PropNet SDE consistently achieves smaller error than other methods in both cases without and with recovery.

In the fourth experiment, we consider the propagation model where activation rates are time-varying instead of constant. More specifically, t_{ij} , the time for node i to activate j , follows the Rayleigh distribution (a specific type of Weibull distribution with $\theta = 2$). For ConTinEst, we generate 1,000 simulations where each one has 15 least labels. For PropNet SDE, we use step size $h = 0.01$ and $L = 1,000$. The results without recovery scenario are shown in Figure 6.4. Mean Field relies on the Markov property of constant activation rates and hence is not capable to handle this situation. For ConTinEst and PropNet SDE, we show their estimated influence $\hat{\mu}(t)$ and the corresponding relative error $|\hat{\mu}(t) - \mu(t)| / \mu(t)$ in the first and second columns of Figure 6.4, respectively. In particular, we observe that both methods are accurate in predicting overall influence, whereas ConTinEst seems more accurate in the early stage, and the proposed PropNet SDE tends to yield smaller error in middle to late stages of the propagation. In addition, we also show the relative accumulated error in individual activation probability, $\sum_i |\hat{x}_i(t) - x_i(t)| / \sum_i x_i(t)$, of the proposed PropNet SDE method

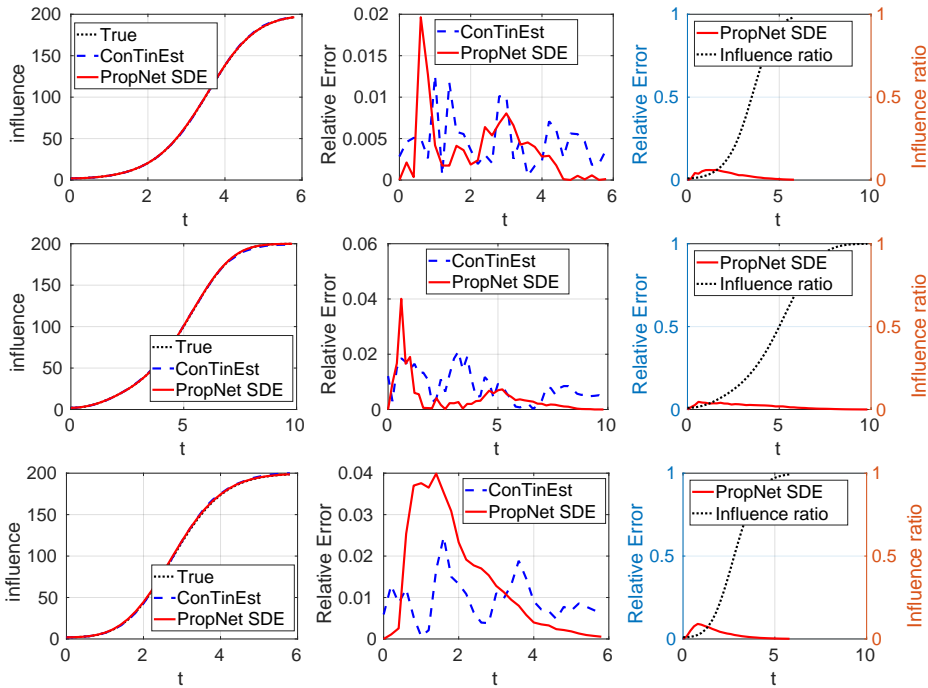


FIG. 6.4. Influence prediction by compared methods on Erdős-Rényi network (top row), small-world network (middle row), and scale-free network (bottom row), all of size $n=200$, using the propagation model with Rayleigh distributed activation times *without* recovery. Left column: True total influence $\mu(t) = \sum_i x_i(t)$ and influences $\hat{\mu}(t) = \sum_i \hat{x}_i(t)$ obtained by ConTinEst and PropNet SDE. Middle column: relative error in influence $|\hat{\mu}(t) - \mu(t)|/\mu(t)$. Right column: relative error in individual activation probability $\sum_i |\hat{x}_i(t) - x_i(t)|/\sum_i x_i(t)$ (influence ratio $\mu(t)/n$ is plotted in black dotted line for reference).

in the right column of Figure 6.4. The influence ratio $\mu(t)/n$ is again plotted in black dotted line for reference. As we can see, PropNet SDE attains very small relative error which indicates that all $x_i(t)$ are estimated accurately. Note that ConTinEst is not capable of computing such estimations. The results are shown in Figure 6.5. We again observe accurate prediction of both total influence $\mu(t)$ and individual activation probability in the middle and right columns of Figure 6.5.

The proposed PropNet SDE algorithm benefits from the variance reduction (VR) technique, which significantly reduces the number of samplings to achieve the same level of accuracy. To demonstrate the effectiveness of variance reduction, we run PropNet SDE with standard sampling of Poisson numbers and VR (i.e., Algorithm 1) using Erdős-Rényi network. The parameter setting of Erdős-Rényi network is the same as above, but we run the algorithms with 5, 10, 20, 40, 80 and 100 sampled cascades to track the prediction errors. We plot both prediction errors with 95% confidence intervals in Figure 6.6. From the plots in Figure 6.6, we can see that PropNet SDE with VR produces lower prediction error than that without VR for the same amount of samplings. This suggests that VR is a simple and effective implementation in the proposed PropNet SDE method to improve computational efficiency. We also show the comparison of Monte-Carlo (MC), ConTinEst, and PropNet SDE with VR on the same dataset in Figure 6.7, where the left and right panels show the prediction errors in influence and individual activation probability respectively.

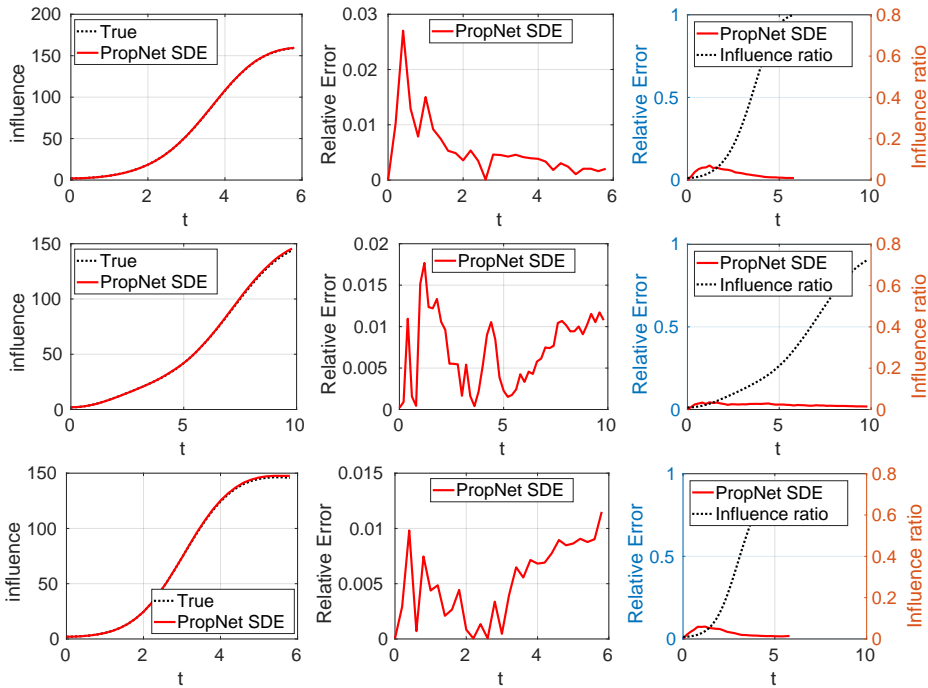


FIG. 6.5. Influence prediction by compared methods on Erdős-Rényi network (top row), small-world network (middle row), and scale-free network (bottom row), all of size $n=200$, using the propagation model with Rayleigh distributed activation times with recovery. Left column: True total influence $\mu(t) = \sum_i x_i(t)$ and influences $\hat{\mu}(t) = \sum_i \hat{x}_i(t)$ obtained by ConTinEst and PropNet SDE. Middle column: relative error in influence $|\hat{\mu}(t) - \mu(t)|/\mu(t)$. Right column: relative error in individual activation probability $\sum_i |\hat{x}_i(t) - x_i(t)|/\sum_i x_i(t)$ (influence ratio $\mu(t)/n$ is plotted in black dotted line for reference).

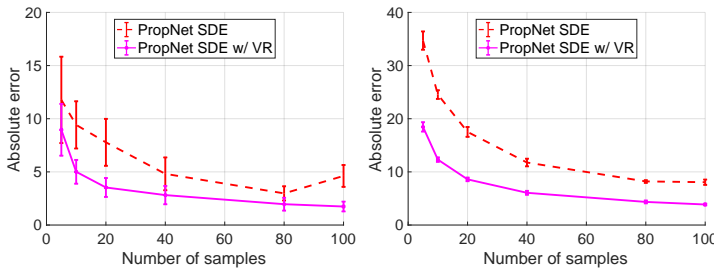


FIG. 6.6. Comparison of prediction errors in influence (left) and individual activation probability (right) versus the number of sampled cascades L using the PropNet SDE without and with variance reduction (VR) for the Erdős-Rényi's network in the first experiment.

6.3. Experiments on real data. We also test the proposed influence prediction algorithm on the Weibo-Net-Tweet dataset¹. The Weibo-Net-Tweet dataset contains 1.7 million users and 400 million edges (followee-follower relationships). This dataset also contains 300,000 popular microblog diffusion episodes (propagation cascades) posted

¹<https://cn.aminer.org/data-sna#Weibo-Net-Tweet>

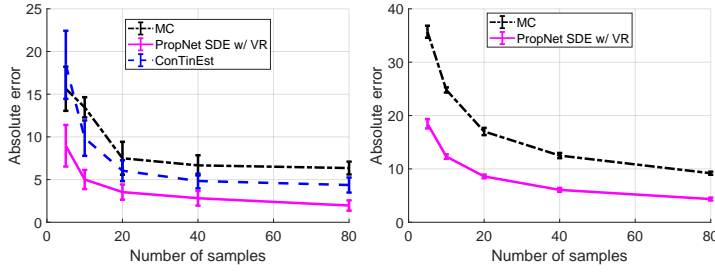


FIG. 6.7. Comparison of prediction errors in influence (left) and individual activation probability (right) versus the number of sampled cascades L using Monte-Carlo simulations (MC), ConTinEst, and PropNet SDE with variance reduction (VR) for the Erdős-Rényi’s network in the first experiment.

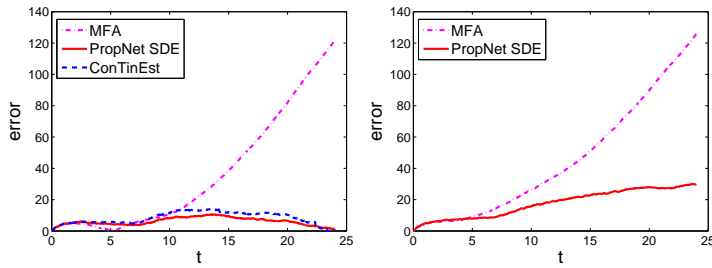


FIG. 6.8. Average error of predicted influence $|\hat{\mu}(t) - \mu(t)|$ (left) and individual activation probability $\sum_i |\hat{x}_i(t) - x_i(t)|$ (right) versus time (in hours) on the Weibo-Net-Tweet dataset for the first 24 hours of propagations.

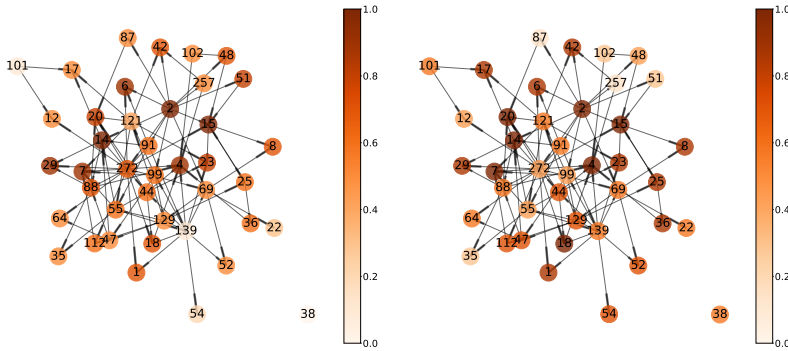


FIG. 6.9. Estimation of individual activation probabilities at $T=24$ hours using the proposed PropNet SDE method (left) and the empirical activation probabilities exhibited by the testing data (right) from source set $S = \{2\}$ on the Weibo-Net-Tweet dataset. Thicker ends of edges indicate receiving sides. Nodes with $<50\%$ activation probabilities in both plots are removed for cleaner looks.

in 2012. Each diffusion episode consists of the original microblog and its retweets. We select the most influential 434 users and retrieve all the propagation cascades that contain at least 50 of these users. Then we randomly select 60% of these cascades, and apply the NetRate algorithm [11] to learn the activation rate matrix A . With the learned A , we apply the comparison algorithms to the source sets of the remaining 40%

cascades to predict influence, and compute the error of the predicted influence to the true influence in the first 24 hours exhibited by these cascades. The results are shown in Figure 6.8. From these plots, we observe Mean Field generates significantly larger errors than ConTinEst and PropNet SDE, where the latter two have comparable accuracy in influence prediction, as shown in the left panel of Figure 6.8. In addition, PropNet SDE also accurately predicted the activation probabilities of individual users as shown in the right panel of Figure 6.8.

We also instantiate one prediction result of individual node activation probabilities starting from a single source node with index 2 (actual user ID is concealed, same below) using the proposed PropNet method. The result is shown in the left panel of Figure 6.9. The color of a node shows the probability that the node is activated at $T = 24$ hours, where the color bar is plotted on the side for reference. For comparison, we also extract the empirical probabilities from the cascades in testing dataset, and show the result on the right panel of Figure 6.9. For cleaner appearance, we removed all nodes that have lower than 50% activation probability in both results. Using the prediction result of the PropNet SDE, we can see that nodes indexed by 4, 7, 8, 18, 20, 23, 29 have relatively high probabilities to be activated (retweet the post), in addition to the active direct followers (e.g., nodes 6, 14, 15, 42) of node 2. On the other hand, some direct followers, such as node 257, of node 2 do not often help to spread the post. These claims are supported by the empirical results in the testing data, as shown on the right of Figure 6.9, and they are also backed up by the small quantitative prediction error we showed in the right panel of Figure 6.8.

7. Conclusions

In this paper, we proposed to reformulate propagations on heterogeneous networks using JSDE system. We also developed an efficient numerical scheme to solve the JSDE system and predict influence. Using a series of numerical experiments on variety of networks and propagation models, we show that our proposed method is accurate and efficient in influence prediction when compared to the state-of-the-art methods. Moreover, we showed that the proposed method can be readily modified and applied to more general propagation models which cannot be handled by any existing methods.

Acknowledgements. GB and YZ are supported in part by NSFC Innovative Group Fund (No.11621101). XY is supported in part by NSF DMS-1620342, CMMI-1745382, DMS-1818886 and DMS-1925263. H. Zha is supported in part by NSFC Grant No. 61672231 and Grant No. U16092202, as well as Shenzhen Research Institute for Big Data. H. Zhou is supported in part by NSF DMS-1620345 and DMS-1830225, and ONR N00014-18-1-2852.

REFERENCES

- [1] Á. Bodó, G.Y. Katona, and P.L. Simon, *SIS epidemic propagation on hypergraphs*, Bull. Math. Biol., **78(4):713–735**, 2016. [1](#), [2](#)
- [2] R.F. Bass, *Stochastic differential equations with jumps*, Probab. Surv., **1:1–19**, 2004. [1](#)
- [3] M. Boguná and R. Pastor-Satorras, *Epidemic spreading in correlated complex networks*, Phys. Rev. E, **66(4):047104**, 2002. [1](#)
- [4] E. Cator and P. Van Mieghem, *Second-order mean-field susceptible-infected-susceptible epidemic threshold*, Phys. Rev. E, **85(5):056111**, 2012. [2](#)
- [5] S.-N. Chow, X. Ye, H. Zha, and H. Zhou, *Influence prediction for continuous-time information propagation on networks*, Netw. Heterog. Media, **13(4):567–583**, 2018. [2](#)
- [6] E. Cohen, *Size-estimation framework with applications to transitive closure and reachability*, J. Comput. Sys. Sci., **55(3):441–453**, 1997. [6.1](#)

- [7] N. Du, Y. Liang, M.-F. Balcan, and L. Song, *Influence function learning in information diffusion networks*, Proc. Int. Conf. Mach. Learn., **2016–2024**, **2014**. **2**
- [8] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, *Scalable influence estimation in continuous-time diffusion networks*, Adv. Neural Inf. Process. Syst., **3147–3155**, **2013**. **1**, **2**, **6.1**
- [9] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, *Multistage campaigning in social networks*, Adv. Neural Inf. Process. Syst., **4718–4726**, **2016**. **1**
- [10] O. Givan, N. Schwartz, A. Cygelberg, and L. Stone, *Predicting epidemic thresholds on complex networks: Limitations of mean-field approaches*, J. Theor. Biol., **288:21–28**, **2011**. **2**
- [11] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, *Inferring networks of diffusion and influence*, ACM Trans. Knowl. Discov. Data, **5(4):21**, **2012**. **6.3**
- [12] M. Gomez Rodriguez, and B. Schölkopf, *Influence maximization in continuous time diffusion networks*, Proc. Int. Conf. Mach. Learn., **1–8**, **2012**. **1**, **2**
- [13] D. Kempe, J. Kleinberg, and É. Tardos, *Maximizing the spread of influence through a social network*, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **137–146**, **2003**. **1**
- [14] D. Kempe, J. Kleinberg, and É. Tardos, *Influential nodes in a diffusion model for social networks*, in L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi and M. Yung (eds.), Automata, Languages and Programming, Springer, **1127–1138**, **2005**. **1**
- [15] J. Liu, P.E. Paré, A. Nedić, C.Y. Tang, C.L. Beck, and T. Başar, *On the analysis of a continuous-time bi-virus model*, Proc. IEEE Conf. Decis. Control, **290–295**, **2016**. **2**
- [16] P.A. Maginnis, M. West, and G.E. Dullerud, *Exact simulation of continuous time Markov jump processes with anticorrelated variance reduced Monte Carlo estimation*, Proc. IEEE Conf. Decis. Control, **3401–3407**, **2014**. **4.2**
- [17] J.C. Miller and I.Z. Kiss, *Epidemic spread in networks: Existing methods and current challenges*, Math. Model. Nat. Phenom., **9(2):4–42**, **2014**. **1**
- [18] H. Narasimhan, D.C. Parkes, and Y. Singer, *Learnability of influence in networks*, Adv. Neural Inf. Proces. Syst., **3186–3194**, **2015**. **2**
- [19] M. Newman, *Networks: An Introduction*, Oxford University Press, **2010**. **1**
- [20] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks*, Rev. Mod. Phys., **87(3):925**, **2015**. **1**, **2**
- [21] E. Platen and N. Bruti-Liberati, *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, Springer Berlin Heidelberg, **2010**. **1**, **4.1**, **4.1**, **4.3**
- [22] F.D. Sahneh and C. Scoglio, *Epidemic spread in human networks*, Proc. IEEE Conf. Decis. Control, **3008–3013**, **2011**. **1**, **2**
- [23] F. D. Sahneh, C. Scoglio, and P. Van Mieghem, *Generalized epidemic mean-field model for spreading processes over multilayer complex networks*, IEEE/ACM Trans. Netw., **21(5):1609–1620**, **2013**. **2**
- [24] K. Scaman, R. Lemonnier, and N. Vayatis, *Anytime influence bounds and the explosive behavior of continuous-time diffusion networks*, Adv. Neural Inf. Proces. Syst., **2026–2034**, **2015**. **2**
- [25] A. Taylor and D.J. Higham, *CONTEST: A controllable test matrix toolbox for MATLAB*, ACM Trans. Math. Softw., **35(4):26**, **2009**. **6.1**
- [26] P. Van Mieghem and J. Omic, *In-homogeneous virus spread in networks*, preprint, [arXiv:1306.2588](https://arxiv.org/abs/1306.2588), **2013**. **2**
- [27] P. Van Mieghem, J. Omic, and R. Kooij, *Virus spread in networks*, IEEE/ACM Trans. Netw., **17(1):1–14**, **2009**. **1**, **2**, **6.1**, **6.1**
- [28] P. Van Mieghem and R. Van de Bovenkamp, *Non-Markovian infection spread dramatically alters the susceptible-infected-susceptible epidemic threshold in networks*, Phys. Rev. Lett., **110(10):108701**, **2013**. **2**
- [29] P. Van Mieghem and R. van de Bovenkamp, *Accuracy criterion for the mean-field approximation in susceptible-infected-susceptible epidemics on networks*, Phys. Rev. E, **91(3):032812**, **2015**. **2**
- [30] M. Vergeer, L. Hermans, and S. Sams, *Online social networks and micro-blogging in political campaigning the exploration of a new campaign tool and a new campaign style*, Party Politics, **19(3):477–501**, **2013**. **1**
- [31] J. Wortman, *Viral marketing and the diffusion of trends on social networks*, Technical Report, **2008**. **1**
- [32] Z. Yang and T. Zhou, *Epidemic spreading in weighted networks: an edge-based mean-field solution*, Phys. Rev. E, **85(5):056106**, **2012**. **2**