Hindawi
Complexity
Volume 2019, Article ID 6902027, 17 pages
https://doi.org/10.1155/2019/6902027

Research Article

Mining the Hidden Link Structure from Distribution Flows for a Spatial Social Network

Yanqiao Zheng,[1] Xiaobing Zhao,[2] Xiaoqi Zhang,[1] Xinyue Ye,[3] and Qiwen Dai[4]

[1]*School of Finance, Zhejiang University of Finance and Economics, China*
[2]*School of Data Science, Zhejiang University of Finance and Economics, China*
[3]*Urban Informatics-Spatial Computing Lab & College of Computing, New Jersey Institute of Technology, USA*
[4]*School of Economics & Management, Guangxi Normal University, China*

Correspondence should be addressed to Xiaoqi Zhang; xiaoqizh@buffalo.edu

Received 30 December 2018; Revised 3 March 2019; Accepted 31 March 2019; Published 2 May 2019

Academic Editor: Giulio Cimini

This study aims at developing a non-(semi-)parametric method to extract the hidden network structure from the $\{0, 1\}$-valued distribution flow data with missing observations on the links between nodes. Such an input data type widely exists in the studies of information propagation process, such as the rumor spreading through social media. In that case, a social network does exist as the media of the spreading process, but its link structure is completely unobservable; therefore, it is important to make inference of the structure (links) of the hidden network. Unlike the previous studies on this topic which only consider abstract networks, we believe that apart from the link structure, different social-economic features and different geographic locations of nodes can also play critical roles in shaping the spreading process, which has to be taken into account. To uncover the hidden link structure and its dependence on the external social-economic features of the node set, a multidimensional spatial social network model is constructed in this study with the spatial dimension large enough to account for all influential social-economic factors. Based on the spatial network, we propose a nonparametric mean-field equation to govern the rumor spreading process and apply the likelihood estimator to make inference of the unknown link structure from the observed rumor distribution flows. Our method turns out easily extendible to cover the class of block networks that are useful in most real applications. The method is tested through simulated data and demonstrated on a data set of rumor spreading on Twitter.

1. Introduction

Flow data has been widely studied by different disciplines [1–6]. Especially in recent years, the development of internet makes an increasing amount of flow data sets publicly available, among them new types of flows are emerging and attracted more and more attentions from scholars [7, 8].

Unlike the physical movement, such as the trajectory of taxi, the information flow data, such as the time series of the retweet status of a class of tweet articles within a population, does not contain any trajectory-level information, because a user may tweet after he saw many friends had done so. In that case, a group of friends can contribute to the spreading of the tweet, and it becomes impossible to figure out which one is the real single source, neither is it possible to track the trajectory of retweeting. Therefore, this flow data are no longer stored as a collection of well-defined trajectories; instead, they consist of a time series of distributions of a given kind of information within entire population. In addition, the distribution flows are highly "context-dependent", which means the social-economic factors behind every agent joining the spreading process (such as the education, income, and the neighborhood) might significantly affect the speed, extent, and coverage of spreading, suggesting a spatial social network to be uncovered from the distribution flows.

Of course, the emergence of new types and new features of flow data inevitably brings unprecedented opportunities to improve our understanding of interaction patterns between people and thus enrich relevant theories, but the missing observation on the trajectory-level information and the add-in of social-economic context make it challenging to uncover the agent-to-agent links, or equivalently the entire hidden

interaction network. As a result, it becomes necessary to develop more data-driven approaches tailored to uncover the hidden spatial social network behind the distribution flows.

Distribution flows are frequently studied in the field of rumor and/or flu spreading. Existing methods, in broad terms, can be suppressed into two classes, the agent-based modelling/simulation/calibration (ABM) techniques [9–15] and the differential equation (DE) based approaches [16–19]. The class of DE approaches is helpful to derive qualitative conclusions regarding the steady state distribution of the spreading processes and how the equilibrium depends on model parameter in a coarse sense. However, to guarantee the meaningful qualitative results are achievable, the setup of differential equations is often oversimplified, but it would cause the loss of insights into the complex reality. In addition, due to the lack of explicit solution in most cases, it is not possible to apply the DE techniques to fit the real data and generate detailed quantitative results. In contrast, ABM approaches are more realistic and suitable for quantitative research on the real distribution flows. However, there are still a couple of shortages in the existing ABM models [20].

First, ABM often assumes that the spreading process is carried on a network where nodes represent agents that can potentially spread out or be infected with a certain type of object (e.g., rumor); edges are the links between agents. Rumor can only be spread between agents linked by edges. Under ABM framework, this interaction network is supposed to be known and prescribed in prior. Prior network may lose critical information of the interaction patterns of population [15, 21, 22]. For instance, in the Twitter network, a natural interaction network structure is the network formed by friendship or followership relation between users which is also frequently used as the prior network for rumor spreading studies [14, 18]. However, rumor does not have to follow this network to spread [23]. In fact, the retweet action of big name users is more likely to be visible through other channels to those users who are not linked in the friendship network, such as by TV shows, and newspapers. Therefore, the spreading between a big name user and an ordinary user is still possible even if they are not linked at all by merely counting the friend or follower relation. The existence of hidden links makes prior network fail to capture all structural features of interactions, a data-driven or posterior network would be helpful to overcome this issue.

Second, given the prior network, ABM assumes the spreading occurs through interaction mechanisms between two randomly picked agents. Widely used interaction mechanisms include the independent cascade model and the linear threshold model and so on [24–26]. These mechanisms are often parametrized and assumed homogeneous for all agents; i.e., the mechanism is determined by a set of parameters that are constant and invariant for different agents. In reality, both of the relative positions of an agent within the network, such as the degree, centrality, betweenness of an agent [17, 18, 27], and many social-economic factors external to the entire network, such as the geographic location, social status, education level, and wealth [22, 27], can drastically affect the likelihood that agents get infected by the rumor. But the heterogeneity among agents is often missing from the standard ABM framework.

To resolve the above issues, we propose a novel and completely data-driven modelling approach to characterize the hidden interaction network and the spreading process. Our study contributes to the existing literature in the following aspects.

First, we consider the interaction network as a weighted multidimensional spatial social network, which is an extension to the standard spatial network and the nodes in the network are embedded into a multidimensional feature space $\mathbb{R}^p$. The weighted edge between nodes is considered as a continuous function on $\mathbb{R}^p \times \mathbb{R}^p$. Within such a network, the value of edge weight function can depend on features of both the start nodes and end nodes, so it gives full respect to the heterogeneity of nodes and its effect on shaping the spreading process and distribution flow.

Second, we link the interaction network with the distribution flows by the classical mean-field models [9, 16–18] and the law of distribution transition is realized by a kernel operator with its kernel function given by the edge weight function. Such a construction allows the infection status of a given node to depend on all other nodes in the network in a smooth manner, which avoids the arbitrariness of distinguishing the impact of neighbor and nonneighbor nodes, while also facilitating the inclusion of the context information embedded in the spatial social network into the analysis of spreading.

Third, we adopt the kernel smoothing technique and nonparametric likelihood estimation from statistics [28, 29] to fit our model into real distribution flows, where the entire edge weight function is supposed to be unknown and needs to be estimated from the distribution flow data from the real world. The nonparametricity makes our method a powerful tool of information mining for distribution flow data. Finally, the widely used block models [30–33] can be easily incorporated into our framework, which helps better uncover hidden social-economic connections between individuals from distribution flows.

The paper is organized as follows. In Section 2, we give an overview of existing methods of network estimation. Section 3 formally presents the setup of our method, including the definition of feature space network, mean-field models and their simulation techniques, and the design of our likelihood estimators. Section 4 validates the effectiveness of our estimators to the hidden network by synthetic data and numerical experiments. Section 5 applies our method to a distribution flow dataset of the information spreading on Twitter relevant to the event "Unite the Right rally", 2017.

2. Relevant Methods

The proposed method in this paper is essentially a network estimation tool, while network estimation is a long-standing topic in many different fields.

In the studies of agent-based model (ABM), simulation-based estimation is usually adopted to calibrate the unknown parameters involved in model setup [13–15, 18, 34, 35]. Simulation-based estimation is efficient in dealing with the

estimation of ABMs as it is often impossible to derive an analytic expression for the standard error functions in ABM setting; simulation can help generate an empirical version of the error function and facilitate the application of standard ordinary least square (OLS) and maximum likelihood (ML) estimation strategy. However, the simulation-based estimation is more frequently applied to parametric ABM where only a finite-dimensional parameter vector is to be estimated; it is rarely used to estimate the hidden network structure as the unknown network is essentially nonparametric, which is less tractable than the parametric models. To our best knowledge, the only exception comes from Grazzini and Richiardi [35]; Kukacka and Barunik [36], in which the interaction mechanism when two agents meet is allowed to include a nonparametric component, and the kernel smoothing method and nonparametric likelihood (or least square) estimators are applied to cope with model estimation. However, Grazzini and Richiardi [35]; Kukacka and Barunik [36] do not include the interaction network between agents into their analysis, nor the model identifiability issue is resolved; thus, further exploration is needed in this direction.

The other related works deal with link prediction by stochastic-network models. In this field, nonparametric tricks are more often adopted to make inference of hidden features of stochastic network [23, 31, 32, 37, 38]. Lü and Zhou [31] review the main-stream heuristic algorithms to forecast the missing links within a partially observed network. Bickel et al. [39], from the perspective of statistic inference, summarize and validate the application of variational expectation maximization (VEM) algorithm to infer the probability of existence of a link between two nodes from observed edge data. Matias et al. [38] extend the VEM method to deal with the future occurrence probability of edges given a dynamic linked network and the historic edge data; this extended method can handle the case where the evolution of occurrence probability depends nonparametrically on an unknown hazard function. All these methods were developed under a common assumption that at least the edge information of part of the network has already been observed, which is possible for trajectory data, but not possible for distribution flows. Thus, a further extension is needed to handle the case that all edge data are missing.

In the literature of physics, the task of detecting the hidden network link structure from node-level time-series data is phrased as "network reconstruction". Taking distribution flows as the input, two outstanding network reconstruction methodologies are directly comparable to ours. One is based on the compressive sensing technique as proposed in Shen et al. [40]; the other is based on the combination of likelihood estimation and the mean-field approximation technique as discussed in Roudi and Hertz [41]. The basic idea in Shen et al. [40] is to convert the network reconstruction problem to a classical convex optimization problem with linear constraints, which is the so-called compressive sensing (CS) problem. In the CS problem, the linear constraints come from the transition probability of nodes within the network from the uninfected state to the infected state, while the objective function arises from the sparsity assumption regarding the network link structure. Unlike the applications of CS approach to the network reconstruction from continuous time-series data [42–44] where the feature variables associated with every node are directly observable, in the case of distribution flows, the key variable, transition probability, is not observable from the data. Therefore it has to be calculated so as to form the required linear constraints. Inferring the transition probability from the $\{0, 1\}$-valued distribution flow data requires a stationary assumption on the underlying model which is too restrictive in many applications. For instance in the spreading of virus, an agent might die immediately after it is infected, in which case the infected agent is censored in the sense that its infectious status is constantly one since the time of being infected. When censored agents exist in the network, stationarity of the transition is impossible and the CS framework in Shen et al. [40] is no longer applicable. The other problem of the CS framework is its incapability of handling the spatial heterogeneity among different nodes. As we have highlighted that the education, wealth, and many other social-economic factors can play critical roles to determine the link strength among people and therefore affect the information spreading dynamics, modelling the dependence of the hidden link structure on those social-economic factors is necessary in the studies of social network. The inclusion of social-economic factors would introduce heterogeneity among nodes, which makes it challenging to identify which two nodes are relatively homogeneous and can be grouped together. In the CS framework, grouping different nodes is the premise to calculate the transition probability. In an abstract network, all nodes are homogeneous, and the grouping can be simply taken as the set of all nodes as done in Shen et al. [40], while in a spatial network with heterogeneity widely existing such a simple grouping trick is meaningless. How to extend the CS framework to spatial social network becomes a tough job and extensive studies are needed.

The deep reason that restricts the CS framework is its reliance on the unobservable transition probability. That restriction can be effectively resolved by applying the likelihood technique as suggested in Roudi and Hertz [41]. The goodness of likelihood-based approach is that it can compute the unknown transition probability simultaneously with the other model parameters. But the computation usually takes too much time because there is no explicit solution for the first-order condition of the maximum likelihood; numerical solution is required. To make the computation easier, a mean-filed approximation technique is presented in Roudi and Hertz [41], which can definitely increase the computation speed. However, the approximation can only work for the case that all link strengths have to be close to zero, which restricts its usefulness in many applications of social network. On the other hand, the current version of the approximation technique in Roudi and Hertz [41] still assumes an abstract network structure, and no dependence of the link strength on social-economic factors is allowed; it is unclear whether the approximation is extendible to account for the reconstruction of spatial social networks. Finally, Roudi and Hertz [41] are only concerned with the situation that the number of nodes (N) is relatively small, and the computation complexity comes mainly from numerically

solving the maximum likelihood problem. But when N is large, the computation complexity would be dominated by the matrix multiplication for the $N \times N$ adjacency matrix. Since the approximation technique in Roudi and Hertz [41] still requires the matrix multiplication, its speed-up effect for giant networks may not be that significant. More explorations on the fast reconstruction of giant spatial social networks are needed.

3. Model Setup

3.1. Feature Space Network. We consider a weighted multidimensional spatial social network, where nodes of the network are considered as elements in a p-dimensional Euclidean space $\mathbb{R}^p$ and every dimension of $\mathbb{R}^p$ is interpreted as a feature of nodes; thus, $\mathbb{R}^p$ is interpretable as a feature space. Edges between nodes are assumed to depend on features of nodes in a smooth way, i.e., edge set of the graph is equivalent to a smooth function (up to a certain order of derivatives) or an almost-everywhere smooth function (i.e., the function is smooth for all points except those contained in a zero-measure set), denoted as $E : \mathbb{R}^p \times \mathbb{R}^p \longrightarrow [0,1]$, where, without loss of generality, edge weight between two nodes is restrained within the unit interval. Such a specification admits a stochastic-network interpretation of our model; the weight can be thought of as the probability that two nodes share an edge. Since the nodes of the network may not be evenly distributed within the entire space $\mathbb{R}^p$, without loss of generality, we assume the node's distribution is characterized by a probability measure F on $\mathbb{R}^p$, and F is supposed to be known from the data. In sum, the p-dimensional spatial network can be recorded as $G(\mathbb{R}^p, E, F)$, or shortly G when there is no ambiguity regarding its nodes space, distribution, and edge function.

There are several advantages to assume that the spreading process and distribution flows occurred within G. First, the embedding of the node set into feature space $\mathbb{R}^p$ allows us to characterize the feature information of nodes that are external to the network structure [21, 22, 27], which are usually as important as the network structure itself in determining the spreading process and distribution flows. Luo et al. [22] argue that including social-economic factors such as the intensity of population gathering in a set of locations can significantly increase the capacity of forecast of illness spreading among residents. Viboud et al. [45] report similar findings. Second, allowing nodes unevenly distributed within the feature space admits us to include more general network into analysis. For instance, by proper choice of the measure F (e.g., finitely supported), it is even possible to consider a network with only finitely many nodes but sitting in the infinite feature space $\mathbb{R}^p$; this allows us to include most of networks that we can meet in practice. Finally, allowing the edge weight to smoothly depend on features of both the flow-in and flow-out nodes makes it possible to incorporate the background information into the interaction mechanism; this is critical when the network itself is only a small component of a larger background system [27]. In addition, a by-product of treating edges as a smooth function is its induced computational efficiency. In fact, when a network consists of a giant number of nodes, even a simple summation operation can take a long time and huge memory, but when edges vary smoothly along with nodes, it becomes possible to only do calculation on a small set of nodes and the global features of edges then can be inferred from the result on the relatively small set by the kernel smoothing technique from nonparametric statistics [28, 29]. Based on these advantages, we will concentrate on the spatial network, $G(\mathbb{R}^p, E, F)$, instead of a more general concept of network.

3.2. Mean-Field Models. To model spreading processes within a spatial network $G(\mathbb{R}^p, E, F)$, we follow the convention in the studies in rumor spreading literature [10, 17] and adopt the common assumption that a rumor can be spread out from a node x to the other y if and only if (1) the initial node x must have been infected with the rumor, recorded as the event $I(x) = 1$; (2) there is an edge between them, or equivalently, $E(x, y) > 0$; and (3) when condition (1) and (2) hold, whether or not the spreading actually happens is purely random up to a probability r. Different spreading models impose different requirement on the probability r. In the current studies, we adopt the mean-field model to determine r, as suggested in most of previous studies. Formally, for every fixed time t, the probability of node $x \in \mathbb{R}^p$ being infected is determined by the following mean-field equation:

$$\frac{dr(x,t)}{dt} = (1 - r(x,t)) \cdot \int_{\mathbb{R}^p} E(x,y)\, r(y,t)\, dF(y) \quad (1)$$

The interpretation of (1) is that at t the temporal variation rate of the probability that node x is infected (represented as $dr(x,t)/dt$) is a proportion to the probability that node x has not yet been infected by time t (represented as $1 - r(x,t)$), and the proportion is determined through a weighted sum of the probability of all other nodes in the network having been infected by t. The weight function describes the strength of connection between nodes x and y, thus can be formulated as the edge function E. Using the classical result of mean-field equations [46–50], it can be easily verified that the infection probability $r(x,t)$ in (1) is exactly equal to the probability of $I(x,t) = 1$ for a given right-continuous mean-field point process I satisfying the following:

$$\begin{aligned} &E\left(I(x,t) - I\left(x,t^-\right) \mid I\left(x,t^-\right) = 0\right) \\ &\quad = \int_{\mathbb{R}^p} E(x,y)\, I\left(y,t^-\right) dF(y) \end{aligned} \quad (2)$$

where $I(x,t^-)$ is the left-limit of process $I(x,\cdot)$. The interpretation of (2) is more straightforward than (1); (2) points out that the average rate of node x being infected is contributed by all those nodes that (1) have a connection to x and (2) have been infected by the current time. These two conditions are often imposed in literature.

Let r be a function satisfying the functional differential equation (1), also denote f as the density or mass function associated with probability F; then, the event that a given node x is observed at time t and its infectious status is observed to be infected has the probability density:

$$\mathfrak{p}_1(x,t) = f(x)\, r(x,t) \quad (3)$$

in contrast, the density for the event that x is observed to be uninfected at t is given as

$$\mathfrak{p}_0(x,t) = f(x)(1 - r(x,t)). \tag{4}$$

Suppose that given a time t, the infectious status of a set of randomly picked nodes $\mathcal{N} \in \mathbb{R}^p$ is observable and represented as

$$\mathcal{O}_t = \{I(x,t) : x \in \mathcal{N}\} \tag{5}$$

with $I(x,t) = 0$ being not infected and $I(x,t) = 1$ being infected; then, the likelihood function of the observations $\mathcal{O}_t$ can be written in the following way by using (3) and (4):

$$L(\mathcal{O}_t, E) = \prod_{x \in \mathcal{N}} \left(f(x) r(x,t)\right)^{I(x,t)} \left(f(x)(1 - r(x,t))\right)^{1 - I(x,t)} \tag{6}$$

where we add the edge function E into likelihood because it affects L through determining the functional form of r. Maximizing (6) can yield the classical maximum likelihood (ML) estimator of E.

3.3. Nonparametric Likelihood Estimator and Kernel Smoothing. In the study of spreading process, only the distribution flows of the form (5) are available; the details of link structure between nodes represented by edge function E are not observable, thus need to be estimated. In this section, we construct a nonparametric simulated maximum likelihood estimator (NPSML) to the functional form of E given the observed distribution flows $\{\mathcal{O}_{t_i} : i = 1, \ldots, T,\ t_1 < \cdots < t_T\}$ on a sequence of time. The NPSML is an efficient nonparametric inference technique proposed by Kristensen and Shin [29]. NPSML applies well to the case where an explicit expression of the likelihood function is not achievable, which is exactly what we need to handle because the distribution function r in (6) is the solution to the functional differential equation (1); there is no clean analytic expression available for it.

However, our task is different from the situation discussed originally in Kristensen and Shin [29]. First, the original NPSML applies nonparametric kernel smoothing to approximate the unknown likelihood function; the model generating the likelihood function is still parametric, but in (6) the likelihood depends on the nonparametric edge function E. To this situation, one extra kernel smoothing step is needed to approximate E. Second, in Kristensen and Shin [29]; Kukacka and Barunik [36], simulation is conducted on the level of random variable, while, in our case, simulation is on the level of distribution that is equivalent to numerically solve the mean-field equation (1). Finally, due to the involvement of nonparametric model setup, the model identifiability has to be checked in order to guarantee the correctness of the resulting estimation.

Due to the first and second differences, we provide the following algorithm to generate the simulated likelihood function (in the following constructions, we always use K^p to denote the p-dimensional standard Gaussian kernel function, $K_h^p(x) = K^p(x/h)/h^p$ for some positive constant h.).

Step 1. Select constant $dt > 0$, large positive integer M_1 and M_2 (dt is the length of every time step used for numerically solving the functional differential equation (1), M_1 and M_2 are the number of random samples that will be drawn to generate the kernel smoothing approximation to the unknown likelihood function and edge weight function).

Step 2. Draw M_1 random samples $x_1, \ldots, x_{M_1} \in \mathbb{R}^p$ from distribution F, and M_2 random samples $w_1, \ldots, w_{M_2} \in \mathbb{R}^p \times \mathbb{R}^p$ from the product measure $F \otimes F$.

Step 3. Given $e_1, \ldots, e_{M_2} \in [0,1]$, construct function $\widehat{E}$ as follows:

$$\widehat{E}(w) = \frac{\sum_{i=1}^{M_2} K_{h_1}^{2p}(w - w_i) \cdot e_i}{\sum_{j=1}^{M_2} K_{h_1}^{2p}(w - w_j)} \tag{7}$$

Step 4. Given t_i, let $\mathcal{O}_{t_i} = \{I(y_1, t_i), \ldots, I(y_M, t_i)\}$ denote the observation set at time t_i whose cardinality is M, constructing function $\widehat{r}(., t_i)$ as follows:

$$\widehat{r}(y, t_i) = \frac{\sum_{l=1}^{M} K_{h_2}^p(y - y_l) \cdot I(y_l, t_i)}{\sum_{j=1}^{M} K_{h_2}^p(y - y_j)} \tag{8}$$

Step 5. Solve mean-field equation (1) over interval $[t_i, t_{i+1})$ at the set of sample point $\{x_1, \ldots, x_{M_1}\}$ drawn in Step 2 by Euler's method with time step dt subject to the initial condition $\widehat{r}(., t_i)$ as follows:

$$\widehat{r}(x_j, t_i + (k+1) \cdot dt) = \widehat{r}(x, t_i + k \cdot dt) + \left(1 - \widehat{r}(x_j, t_i + k \cdot dt)\right) \cdot \frac{dt}{M_1} \cdot \sum_{l=1}^{M_1} \widehat{E}(x_j, x_l)\, \widehat{r}(x_l, t_i + k \cdot dt) \tag{9}$$

where $k = 0, 1, \ldots, \lfloor (t_{i+1} - t_i)/dt \rfloor$, $\lfloor a \rfloor$ is the greatest integer less than a.

Step 6. For the observation set $\mathcal{O}_{t_{i+1}} = \{I(y_1, t_{i+1}), \ldots, I(y_{M'}, t_{i+1})\}$ at t_{i+1} with cardinality M', generate the simulated density at the sample nodes $\{y_l : l = 1, \ldots, M'\}$ as follows:

$$\widehat{r}(y_l, t_{i+1}) = \frac{\sum_{j=1}^{M_1} K_{h_3}^p(y_l - x_j) \cdot \widehat{r}(x_j, t_{i+1})}{\sum_{j=1}^{M_1} K_{h_3}^p(y_l - x_j)} \tag{10}$$

and construct the simulated likelihood function as follows:

$$\widehat{L}\left(\mathcal{O}_{t_{i+1}}, \{e_1, \ldots, e_{M_1}\}\right) = \prod_{l=1}^{M'} \left(f(y_l)\, \widehat{r}(y_l, t_{i+1})\right)^{I(y_l, t_{i+1})} \cdot \left(f(y_l)(1 - \widehat{r}(y_l, t_{i+1}))\right)^{1 - I(y_l, t_{i+1})}. \tag{11}$$

The full information likelihood function for all observation time can be constructed from (11) in the following way:

$$\widehat{L}^{*}\left(\left\{\mathcal{O}_{t_i} : i = 1, \ldots, T\right\}, \left\{e_1, \ldots, e_{M_2}\right\}\right) = \prod_{i=1}^{T} \widehat{L}\left(\mathcal{O}_{t_i}, \left\{e_1, \ldots, e_{M_2}\right\}\right) \tag{12}$$

The estimator of unknown edge function E can be derived from maximizing the simulated full information likelihood function (12) by selecting appropriate $\{e_1, \ldots, e_{M_2}\}$; the final estimator $\widehat{E}^{*}$ is constructed from the optimal $\{e_1^{*}, \ldots, e_{M_2}^{*}\}$ in the way of (7).

Comparing to NPSML in Kristensen and Shin [29], the algorithm in our study includes one extra sampling step to draw M_2 random points from $\mathbb{R}^p \times \mathbb{R}^p$ which are used for approximating unknown E. In addition, there are two kernel smoothing steps (Steps 4 and 6) regarding the density function r, one for the initial density in the starting time t_i and the other for the end-time density at t_{i+1}. The two kernel smoothing steps are not required when the total number of nodes are small (a few hundred or a few thousand), in which case the whole set of nodes is directly used as the M_1 samples drawn in Step 2. However, when the system has a giant node set (say millions), the sample size $M_1 \ll M$ can be applied in order to lift the computation efficiency. Moreover, the node sets being observed at different observation time may not always be identical; it is more often the case that when a node is tracked to be uninfected at some time t, it will be regarded as safe and missing from the consecutive tracking in the next few observation time points. In this interval-censor situation, the M_1 sampled nodes and the two kernel smoothing steps are needed to avoid the noise induced by censoring.

As documented in Kristensen and Shin [29]; Kukacka and Barunik [36], the NPSML estimator does not suffer from the "curse of dimension" despite its nonparametric essence because the number of simulation samples is independent from the number of observation samples. When the latter is large, the inefficiency induced by kernel smoothing vanishes during the aggregation involved in the likelihood function. By the same argument and the fact that in most real-world applications the number of observed nodes is giant, our modified NPSML estimator is free from the curse of dimensionality as well.

3.4. A Fast Algorithm. As shown in (9), the estimation procedure requires repeated evaluation of the multiplication between a $M_1 \times M_1$ matrix and a M_1 dimensional vector; the computation complexity is of the order M_1^2. Although M_1 can be taken as much smaller than the number of nodes in observations (M), it still has to increase as M increases. So when M is a giant number, M_1 has to be large as well; the computation complexity of the entire estimation procedure will be dominated by M_1^2. In this section, we propose a fast algorithm which can reduce the computation complexity in (9) to be linearly dependent on M_1 that is reasonable and implementable in practice.

The idea of the fast algorithm comes from the technique of agent-based simulation (ABS). In every iteration of ABS, every agent in the network is only required to interact with another agent randomly picked from its neighbor. In our setting, there is no strict "neighbor" defined, while it is still possible to randomly pick one agent from the entire population and the interaction is only counted on the given agent and its randomly picked partner. Formally, Step 5 in previous paragraph is split to three substeps.

Step 5(1). For fixed t and fixed $x_j \in \{x_1, \ldots, x_{M_1}\}$, randomly pick one $x_l(j,t)$ from $\{x_1, \ldots, x_{M_1}\}$;

Step 5(2). Compute

$$\widehat{r}\left(x_j, t + dt\right) = \widehat{r}\left(x_j, t\right) + \left(1 - \widehat{r}\left(x_j, t\right)\right) \cdot \widehat{E}\left(x_j, x_l\left(j,t\right)\right) \widehat{r}\left(x_l\left(j,t\right), t\right) dt \tag{13}$$

Step 5(3). Repeat the above two steps for all $t = t_k$, $k = 0, 1, \ldots, \lfloor (t_{i+1} - t_i)/dt \rfloor - 1$, and for all t_is.

Comparing (9) and (13), the main difference is that the inner product of vectors (i.e., the sum over $\{x_1, \ldots, x_{M_1}\}$) is replaced with a scalar multiple, so the resulting computation complexity for all M_1 nodes linearly depends on M_1 which is significantly faster than the original algorithm.

For the accuracy of the fast algorithm, we claim that compared to the original algorithm, the accuracy loss induced by the fastness is controlled by a constant multiple of $\Delta t = \max\{t_{i+1} - t_i : \text{ for all } i\}$. In fact, due to the randomness of $x_l(j,t)$s, it is easily to verify the following:

(i) the expectation of the left hand side of (9) is identical to the expectation of left hand side of (13);

(ii) denote $\Delta(j,t)$ as the increment; $\Delta(j,t) = (1 - \widehat{r}(x_j,t)) \cdot \widehat{E}(x_j, x_l(j,t))\widehat{r}(x_l(j,t),t)$, then for $t_i \le t, t' \le t_{i+1}$, $1 \le j, j' \le M_1$ and all t_is, $\operatorname{cov}(\Delta(j,t), \Delta(j',t') \mid \widehat{r}(x_j,t_i)) \le t_{i+1} - t_i$.

The property (i) and the identity for $j' \neq j$ in (ii) are quite trivial. For $t_i < t < t' < t_{i+1}$, then $\operatorname{cov}(\Delta(j,t), \Delta(j,t') \mid \widehat{r}(x_j,t_i))$ can be decomposed as the sum of the following two components

$$\begin{aligned} A &= \operatorname{cov}\left(\Delta\left(j,t\right), \left(1 - \widehat{r}\left(x_j,t\right)\right) \cdot \widehat{E}\left(x_j, x_l\left(j,t'\right)\right) \cdot \widehat{r}\left(x_l\left(j,t'\right), t'\right) \mid \widehat{r}\left(x_j,t_i\right)\right) \\ &= \operatorname{var}\left(\widehat{r}\left(x_j,t\right) \mid \widehat{r}\left(x_j,t_i\right)\right) \cdot E\left(\widehat{E}\left(x_j, x_l\left(j,t\right)\right) \widehat{r}\left(x_l\left(j,t\right), t\right)\right) \\ &\quad \cdot E\left(\widehat{E}\left(x_j, x_l\left(j,t'\right)\right) \widehat{r}\left(x_l\left(j,t'\right), t'\right)\right) \\ &\le \operatorname{var}\left(\widehat{r}\left(x_j,t\right) \mid \widehat{r}\left(x_j,t_i\right)\right) = \operatorname{var}\left(\widehat{r}\left(x_j,t\right) - \widehat{r}\left(x_j,t_i\right) \mid \widehat{r}\left(x_j,t_i\right)\right) \le \left\| \frac{dr\left(x_j,.\right)}{dt} \right\|_{\infty} \left(t - t_i\right)^2 \\ &\le \left(t_{i+1} - t_i\right)^2 \end{aligned}$$

$$
\begin{aligned}
B &= \operatorname{cov}\left(\Delta(j,t), \left(\widehat{r}\left(x_j,t'\right) - \widehat{r}\left(x_j,t\right)\right)\right.\\
&\quad \cdot \widehat{E}\left(x_j, x_l\left(j,t'\right)\right) \cdot \widehat{r}\left(x_l\left(j,t'\right),t'\right) \mid \widehat{r}\left(x_j,t_i\right)\Big)\\
&= \operatorname{cov}\left(1 - \widehat{r}\left(x_j,t\right), \widehat{r}\left(x_j,t'\right)\right.\\
&\quad - \widehat{r}\left(x_j,t\right) \mid \widehat{r}\left(x_j,t_i\right)\Big)\\
&\quad \cdot E\left(\widehat{E}\left(x_j, x_l\left(j,t\right)\right)\widehat{r}\left(x_l\left(j,t\right),t\right)\right)\\
&\quad \cdot E\left(\widehat{E}\left(x_j, x_l\left(j,t'\right)\right)\widehat{r}\left(x_l\left(j,t'\right),t'\right)\right) \le \operatorname{cov}\left(1\right.\\
&\quad - \widehat{r}\left(x_j,t\right), \widehat{r}\left(x_j,t'\right) - \widehat{r}\left(x_j,t\right) \mid \widehat{r}\left(x_j,t_i\right)\Big)\\
&\le E\left(\left|\widehat{r}\left(x_j,t'\right) - \widehat{r}\left(x_j,t\right)\right| \mid \widehat{r}\left(x_j,t_i\right)\right)\\
&\le \left\| \frac{dr\left(x_j,.\right)}{dt} \right\|_{\infty} \left(t - t_i\right) \le \left(t_{i+1} - t_i\right)
\end{aligned}
\tag{14}
$$

where $\|\cdot\|_{\infty}$ is the L^{∞} norm of a bounded valued function. The above inequality holds straightforwardly from the fact $\widehat{r}$ is bounded by 1 and its temporal derivative is given by (1) which is also uniformly bounded by 1; then, the statement (ii) follows immediately.

Using Property (i), (ii) and the law of large number, it is straightforward that the difference between the likelihood function constructed from (9) and by (13) is bounded by a constant multiple of Δt as the number of nodes $M \longrightarrow \infty$. If we further require $\Delta t \longrightarrow 0$ along with $M \longrightarrow \infty$, the two types of calculation of the likelihood function would be asymptotically identical, which leads to the same estimator to the hidden network.

Also notice that by the fast algorithm, the choice of dt is independent with the estimation accuracy, so in practice it can be selected directly as $t_{i+1} - t_i$ to increase the speed.

3.5. Block Network. The NPSML algorithm constructed in previous section can be further extended to make inference for the block network model. As in many applications [33, 38, 39], the existence of connection between two agents is only relevant to the groups they belong to, and the features of agents only affect which group they are assigned to. Without loss of generality, the set of Q groups can be considered as a partition of the set of all nodes; then, the edge function can be decomposed as two components:

(i) the group weight function $E^1 : \mathbb{R}^p \longrightarrow [0,1]^Q$;

(ii) the group-level edge weight E^2, which is a $Q \times Q$ matrix with each entry valued in $[0,1]$;

The edge function E for the block network model can be recovered from (i) and (ii) as follows:

$$E\left(x,y\right) = E^1\left(x\right)^{\top} E^2 E^1\left(y\right) \tag{15}$$

where the image of E^1 is viewed as a Q-dimensional columns vector and the subscript $^{\top}$ represents vector transpose. The group weight function is required to satisfy that for every x and $E^1(x) = (s_1, \ldots, s_Q)$, there exist only one $i \in \{1, \ldots, Q\}$ with $s_i > 0$, which means every node can only have positive probability to belong to at most one group which guarantees the requirement that groups constitute a partition of the node set.

The estimation of block network is equivalent to the estimation of (1) the group weight function E^1, which is unknown and consists of the fully nonparametric component of the network, and (2) the interaction matrix E^2, which is the parametric component of the network. So the estimation is essentially semiparametric. The six-step algorithm discussed in Section 3.3 and the fast algorithm in Section 3.4 are still applicable to that case. The only modification is for Step 3, where the kernel smoothing method is no longer applied to the unknown edge weight E. Instead, it is applied to generate the estimate to group weight E^1. Then the hidden weight function E is constructed from the kernel smoothed E^1 and the given interaction matrix E^2 in the way of (15).

Block network model has many advantages. For instance, when the number of groups involved is small and does not depend on the number of nodes, the number of parameters to solve is only $M_1 Q + Q^2$, while the number is M_2 when there is no block structure at all. To generate good approximation to the true edge function, M_2 has to increase along with the number M_1^2 (although slowly); when the node number in observation is giant, M_1 has to be large as well; then $M_2 \gg M_1 Q + Q^2$. Through block network, we can sharply reduce the dimension of parameter space when solving the maximum likelihood problem, which can significantly lift the computation efficiency.

In addition, block network is much easier to identify than the general fully nonparametric networks, which will be discussed in the next section. Finally, under block network, the equilibrium infectious distribution of the spreading process has a clear analytic expression as stated in the following proposition (proof for Proposition 1 is quite trivial, hence omitted).

Proposition 1. *Denote $E_i^1(x)$ as the projection of vector $E^1(x)$ to its ith coordinate. Define $\mathcal{G}_i = \{x \in \mathbb{R}^p : E_i^1(x) > 0\}$ that consists of the set of nodes belonging to group i; then, within a mean-field model of the form (2) with edge function E given by (15), every equilibrium infection distribution $r(x)$ (i.e., satisfying $(1 - r(x)) \cdot \int_{\mathbb{R}}^{p} E(x,y) r(y) dF(y) \equiv 0$) must have the following form:*

$$
r\left(x\right) = \begin{cases} 0 & \text{if } x \in \mathcal{G}_i,\ \mathcal{P}_i\left(E^2\right)^n r\left(y,t_0\right) \equiv 0 \text{ for all } y, n > 0 \\ 1 & \text{else} \end{cases}
\tag{16}
$$

where $r(y,t_0)$ is the prescribed initial distribution of infectious status, $\mathcal{P}_i$ is the projection of a vector to its ith dimension, and $(E^2)^n$ denotes the nth power of matrix E^2.

Proposition 1 is meaningful in the sense that it links the types of equilibria infectious distribution with the matrix

algebra, facilitating the qualitative analysis of the equilibria distribution. For instance, when E^2 is an upper triangle matrix with all its lower off-diagonal entries being zero and all diagonal and upper off-diagonal entries being strictly positive, such as in (17),

$$\begin{pmatrix} x & x & x & \cdots & x \\ 0 & x & x & \ddots & \vdots \\ 0 & \cdots & x & \cdots & x \\ \vdots & \ddots & 0 & x & x \\ 0 & \cdots & 0 & 0 & x \end{pmatrix} \tag{17}$$

then the equilibrium distribution r and the initial distribution $r(.,t_0)$ satisfy the relation:

$$\begin{aligned} r(x) = 1 \quad \text{iff } x \in \bigcup_{i=1}^{Q'} \mathscr{G}_i \Longleftrightarrow \\ r(x,t_0) > 0 \quad \text{iff } x \in \bigcup_{i=Q'+1}^{Q} \mathscr{G}_i \end{aligned} \tag{18}$$

3.6. Validity of NPSML. Due to the nonparametric nature of the edge function E, its identifiability is tricky. When the spreading process can be observed for multiple times (m times) with random initializations and m is large as assumed in Roudi and Hertz [41]; Shen et al. [40], both of the fully nonparametric network E and the block network (E^1, E^2) are identifiable. However, in real applications, a spreading process can at most be observed for a few times; it is not expected that m can be very large. In that case, the fully nonparametric edge function E is no longer fully identifiable; i.e., there exists $E \neq E'$ that leads to the same likelihood function (6) in the limit case. However, it can be shown that E is identifiable up to compact convex set; i.e., the set $\mathscr{S}_{E_0,r(.,t_0)}\{E : L(\mathscr{O}_t, E) = L(\mathscr{O}_t, E_0)\}$ is a compact convex set within the function space $L_2(\mathbb{R}^p \times \mathbb{R}^p)$, where E_0 stands for the true value of edge function. It can also be proved that the set $\mathscr{S}_{E_0,r(.,t_0)}$ also varies along with the initial infectious status $r(.,t_0)$. Formally, we have that $E \in \mathscr{S}_{E_0,r(.,t_0)}$, if and only if the following holds for all $n = 1, \ldots$:

$$\left(\mathscr{M}_{1-r(.,t_0)}\mathscr{K}_E\right)^n r(.,t_0) \equiv \left(\mathscr{M}_{1-r(.,t_0)}\mathscr{K}_{E_0}\right)^n r(.,t_0) \tag{19}$$

where $\mathscr{K}_E$ is a bounded operator over the functional space $L_2(\mathbb{R}^p$ defined through E as $(\mathscr{K}_E g)(x) := \int_{\mathbb{R}^p} E(x, y)g(y)dF(y)$ for every $g \in L_2(\mathbb{R}^p)$ with F being the default node distribution; $\mathscr{M}_f$ is the multiplicative operator determined by f such that $(\mathscr{M}_f g)(x) = f(x) \cdot g(x)$; the nth power in (19) represents the self-composition of an operator for n times. (19) implies that the identifiability of the true edge function E_0 is limited by the extent of the ergodicity of the spreading process within the node space $\mathbb{R}^p$. For instance, when there exists a small open set $U \subset \mathbb{R}^p$ such that all nodes $x \in U$ are infected before the initial time t_0, i.e., $r(x,t_0) \equiv 1$ for all $x \in U$, then it can be verified by (19) that all functions E that deviate from E_0 only within the band set $U \times \mathbb{R}^p$ are contained in $\mathscr{S}_{E_0}$. On the other hand, if there exists open $U' \subset \mathbb{R}^p$ such that $(\mathscr{M}_{1-r(.,t_0)}\mathscr{K}_{E_0})^n r(x,t_0) \equiv 0$ for all $x \in U'$ and all n, then all functions E that deviate from E_0 only within $U' \times U'$ are contained in $\mathscr{S}_{E_0,r(.,t_0)}$. In both of the two cases, nodes in U or U' are not in the ergodic range of the spreading process; hence, the transmission of their infectious status is not observable. For nodes in U, their infections occur ahead of the observation period hence not observable after the start of spreading, while for nodes in U' it can be verified that they will never be infected over the entire spreading process. Therefore, the identifiability of E_0 is restricted by the experience of the spreading process, which is reasonable.

It is still an open question what conditions added to E_0 and/or $r(.,t_0)$ can guarantee the identifiability of the fully nonparametric E_0. But in the special case of block networks, one simple identifiability condition can be figured out. In fact, for block networks, it is straightforward that (E_0^1, E_0^2) is identifiable, if and only if there does not exist a (E^1, E^2) pair that differs from the true (E_0^1, E_0^2) but leads to the same likelihood function (6) in the limit case, if and only if, for the true E_0^2, the vector space spanned by the family of vectors $\{v_t : t \geq t_0\}$ is the entire feature space $\mathbb{R}^Q$, i.e., $\{v_t : t \geq t_0\}$ has full rank. Q is the number of blocks, $v_t = (v_{t,1}, \ldots, v_{t,Q})^\top$ is a Q-dimensional column vector for every t and for each $q = 1, \ldots, Q$, $v_{t,q} = \int_{\mathbb{R}^p} E_{0,q}^1(x)r(x,t)dF(x)$, $E_{0,q}^1$ is the qth entry of $E_0^1(x)$. To reach the full rank condition, the well-known Wronskian determinant [51] can be applied leading to the following clean-form identifiability condition:

$$\begin{aligned} \det\Big\{v_{t_0}, \operatorname{diag}\left(c - v_{t_0}\right) E_0^2 v_{t_0}, \ldots, \left(\operatorname{diag}\left(c - v_{t_0}\right) E_0^2\right)^{Q-1} \\ \cdot v_{t_0}\Big\} \neq 0 \end{aligned} \tag{20}$$

where c is the other Q-dimensional column vector $(c_1, \ldots, c_Q)^\top$ determined by the true E_0^1 function such that $c_q = \int_{\mathbb{R}^p} E_{0,q}^1(x)dF(x)$ for $q = 1, \ldots, Q$; diag is the operation that convert a Q-dimensional vector to a $Q \times Q$ matrix with its diagonal elements being the given vector. By the polynomial nature of the determinant function, it can be verified that (20) holds "generically" in the sense that the set of E^2s that forces (20) to be constantly equal to 0 is contained in an $Q \times Q - 1$ dimensional surface within $[0,1]^{Q \times Q}$, and for those E^2s that (20) is not constantly 0, the set of v_{t_0} that forces (20) to be 0 is only contained in a $Q - 1$ dimensional surface within $[0,1]^Q$. Therefore, (20) holds for almost all E^2 and v_{t_0} except for some extreme cases that have measure 0 under the standard Lebesgue measure.

The "almost" identifiability for block networks guarantees that in most cases when the number of observed nodes is large and the distribution of observation time is dense, the estimated E^1 and E^2 from the NPSML asymptotically converge to their true values and point-wisely follow multivariate normal distributions. This asymptotic result follows straightforwardly from Kristensen and Shin [29]; Kukacka

and Barunik [36] and the general properties of maximum likelihood estimator. So, the theoretical validity of the estimators developed in previous sections is established.

Remark 2 (sparsity). Although in general the complete identifiability for both the general network and the block network is hard to achieve, but if we follow the idea in the network reconstruction literature, Shen et al. [40] only concentrate on the case that the hidden network is as sparse as possible in the sense: the L^2 norm of the edge weight function $\|E\|_2^2 = \int_{\mathbb{R}^p \times \mathbb{R}^p} (E(x,y))^2 dF(x) dF(y)$ for the general network and/or the entry-wise square sum of the block network $\|E^2\|_2^2 = \sum_{i,j} (E_{i,j}^2)^2$ (this is the L^2 norm on the discrete set with cardinality Q^2) is as small as possible. To automate the selection of the sparsest network, we can consider the L^2 norm function as a penalty and subtract it from the log-likelihood function (6), and then optimizing (6) would guarantee the solution converging to the sparsest network. It is easily verified that such a sparse solution is always asymptotically unique, because as we discussed in previous paragraphs, all networks that can lead to exactly the same log-likelihood function form a compact convex set in the functional space; by the compactness and convexity, there always exists a unique E (or E^2) such that its L^2-distance to the origin reaches the minimum.

4. Numerical Experiment with Synthetic Data

Two synthetic data sets are generated from simulation to test the effectiveness of the NPSML estimator designed in previous sections, one for the fully nonparametric network and the other for the block network. For both examples, the node set $\mathcal{N}$ consists of 200 nodes which are drawn purely randomly from the unit cube $[0,1)^2$; thus, these nodes follow the uniform distribution. Consider the following model setup.

Example 1 (full nonparametric network). Edge function E is negatively proportional to the standard Euclidean distance between two nodes, i.e.,

$$E(x,y) = 1 - \sqrt{\frac{\langle x-y, x-y \rangle}{2}} \tag{21}$$

Example 2 (block network). Set $Q = 3$, block membership function E^1 satisfies

$$E^1(x,y) = \begin{cases} (1,0,0) & if \ \dfrac{x+y}{2} < \dfrac{1}{3}; \\ (0,1,0) & if \ \dfrac{1}{3} \geq \dfrac{x+y}{2} < \dfrac{2}{3}; \\ (0,0,1) & else. \end{cases} \tag{22}$$

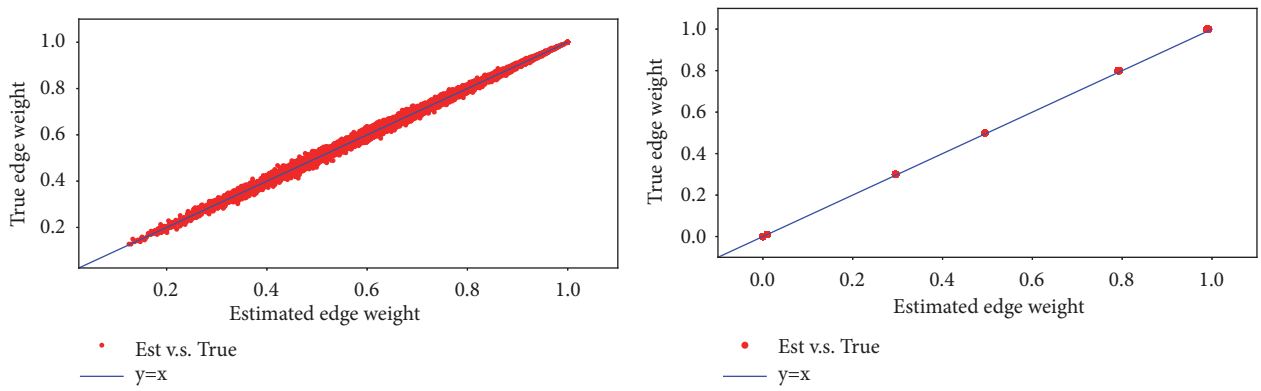
Matrix E^2 is given as follows:

$$E^2 = \begin{pmatrix} 0 & 1 & 0.5 \\ 0.8 & 0 & 0.3 \\ 0.01 & 0 & 0 \end{pmatrix} \tag{23}$$

For both examples, the spreading process is initialized as that 30% of all nodes are infected at the very beginning and the infected nodes are randomly picked from the node set. The full spreading process is generated from a discrete version of (2) with sufficiently small time step (e.g., $dt = 0.01$ that makes the resulting distribution flows as the first-order approximation to the true flows); a coarse time step ($dt = 0.1$) is used for the estimation procedure (9) in order to test the robustness. The process is followed up until day 5; i.e., the time horizon in this simulation study is $[0,t)$ with $t = 5$. The observation of the distribution flows is supposed to be available only at the initial time and the end of every day; i.e., there are 6 chances to observe the distribution of infections at $t = 0, 1, 2, 3, 4, 5$.

For the fully nonparametric Example 1, the spreading process is regenerated for 100 times with 100 random initializations; this is necessary to address the identification issues as pointed out in Section 3.6. For the 100 trails, both the node set and the initial infectious subset are regenerated although their distributions are held constant. For the block network Example 2, the spreading process is generated only once in order to evaluate the fitting performance under the situation that no repeated observation of the spreading process is available. For both examples, the estimated edge function is evaluated on a fixed set of grids for easy comparison where the grid set forms a lattice of the unit cube, i.e., $\mathcal{G} = \{(0.1k, 0.1l) : k, l = 0, 1, \ldots, 10\}$.

If all nodes are included in the computation of the NPSML estimator, there are in principle a 40,000(= 200×200)-dimensional parameter space for full nonparametric network Example 1 and a 609(= $200 \times 3 + 3 \times 3$)-dimensional parameter space for block network Example 2 to be searched which are too time consuming. As in the introduction of NPSML estimator, by the smoothness of edge function, the number of nodes actually used to evaluate the edge function can be much smaller than the size of the entire node set. So, to reduce computation load, we generate another $M_1 = 20$ nodes from the uniform distribution which will be used in Step 3 (Section 3.3) for simulating the distribution function r. Accordingly, the $M_2 = 400$ node pairs will be selected as the product of the 20 nodes for the fully nonparametric Example 1; then, there are 400 parameters to optimize in Example 1, and the size is quite reasonable for most nonparametric tasks. For the block network Example 2, as no node pairs are needed for block networks, there are only 69(= $20 \times 3 + 3 \times 3$) parameters to optimize. As for the selection of kernel width h_1, h_2 and h_3, we set $h_1 = 400^{-1/5}$, $h_2 = 200^{-1/3}$, and $h_3 = 20^{-1/3}$. This is because the kernel smooth method requires kernel width h to satisfy $nh^k \longrightarrow \infty$ and $nh^{k+2} \longrightarrow 0$ in order to guarantee the consistency and asymptotic normality [28, 29, 36, 52], where n is input sample size and k is the dimension of the data. By a rule of thumb, we select the kernel width as $h = n^{-1/(k+1)}$. For h_1, it is only used in Example 1 to estimate the edge function, where the sample size is $M_2 = 400$ and the data dimension is two times of the dimension of node space; thus, k is 4. For h_2 and h_3, they are used in both examples for estimating the distribution function r; thus, data dimension k is always 2. The sample size



(a) Fitting accuracy for fully nonparametric network

(b) Fitting accuracy for block network

FIGURE 1: Fitting accuracy for networks in Examples 1 and 2.

for h_2 is 200 because it is used to turn the real observed r on 200 nodes to its kernel smooth version and the sample size for h_3 is 20 because it turns the estimated r on 20 sampled nodes to its values on the full node set.

For the inference of the block network, the number of block Q is usually not known in prior, so it is also a parameter to estimate. As Q determines the model dimension, we adopt the classical Bayesian information criteria (BIC) introduced in Schwarz [53] to detect the correct model dimension. As defined in Schwarz [53], the greater BIC for a fitted model implies the better explanatory power [53]; therefore, the best choice of Q corresponds to the maximal BIC. In practice, it is not possible to calculate the BIC value for all positive Q, so we follow the convention and only compute the BIC on a small set of $Q \in \{1, 2, 3, 4, 5\}$. The Q associated with the maximal BIC and the corresponding estimates of E^1, E^2 are selected as the final estimators and reported in the following. In our example, the correct $Q = 3$ is always achieved, so we omit this trivial result.

In Figure 1, we plot the difference between the real edge function and the NPSML estimated edge function on the set $\mathcal{G} \times \mathcal{G}$ of node pairs for both examples, where the horizontal axis represents the true value of edge weight on every node pair and the vertical axis represents the estimated weight on the same node pair. To facilitate visualization, Figure 1 is sorted according to the horizontal axis in an ascending manner. The red dots represent the pairs of (estimated weight, true weight), the blue line sketches the identity function $y = x$; therefore, a red dot being closer to the blue line means the better fitting accuracy. Apparently, for most of node pairs, the difference is negligible. To further verify this visual judgement, χ^2 test is carried out for every node pair $(x, y) \in \mathcal{G} \times \mathcal{G}$ with the null hypothesis $E_{x,y} = (E(x, y) - \widehat{E}(x, y))^2 = 0$. Following the asymptotic normality of NPSML estimator $\widehat{E}$ at every (x, y), the distribution of test statistics $E_{x,y}/\sigma^2_{x,y}$ under null hypothesis should be a χ^2 distribution with degree of freedom 1, where $\sigma_{x,y}$ is the asymptotic variance of estimator $\widehat{E}(x, y)$, which can be calculated by bootstrap method. We count the number of node pairs that fail to support the null hypothesis at 90% credential level; the result shows that, in both examples, only less than 10% out of all 10,000 evaluation pairs in $\mathcal{G} \times \mathcal{G}$ fail to support the null hypothesis. So our estimation accuracy is quite satisfactory, which agrees with the visualization in Figure 1.

TABLE 1: Estimation accuracy of E^2.

Entries	Bias	Std.	P value
$E^2_{1,1}$	0.021	0.032	0.468
$E^2_{1,2}$	-0.006	0.012	0.383
$E^2_{1,3}$	-0.003	0.029	0.057
$E^2_{2,1}$	-0.001	0.029	0.028
$E^2_{2,2}$	0.022	0.022	0.66
$E^2_{2,3}$	-0.002	0.028	0.059
$E^2_{3,1}$	0.005	0.024	0.165
$E^2_{3,2}$	0.018	0.029	0.48
$E^2_{3,3}$	0.016	0.021	0.554

For the block network Example 2, Table 1 presents the entry-wise accuracy of estimated E^2 relative to (23), the first column presents the estimation bias, the second and third columns are the empirical standard deviation and the empirical P-values of the estimates, from which we can conclude that the fitting accuracy is relatively perfect.

For robustness check, we also consider the synthetic data generated for different $dt \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$ and the implementation of NPSML estimation on node samples with different size M_1 and M_2. When M_1 and M_2 are increased to 100 and 10,000, respectively, no significant difference can be detected in terms of the estimation accuracy measured by the entry-wise bias between the true and the estimated edge weight, so we omit to plot this result. For the rejection ratio at 90% credential level of the null hypothesis that the true and estimated edge weight are identical, this ratio is lowered down a bit for the block network to less than 6%, but no significant decreasing can be detected for the general network example. This observation might be caused by the fact that for general network, there are much more free parameters to estimate, which reduces the convergence speed. As for the different dt, the variation of estimation accuracy is not significant in

all aspects; this fact agrees with the discussion in the end of Section 3.4.

5. Experiment with Rumor Spreading on Twitter

To demonstrate the usefulness of the NPSML method in real-world applications, we carry out an experiment with the distribution flow data of a real rumor spreading process on Twitter. We collect a data set of tweet articles with regard to the famous event "Unite the Right rally". The "Unite the Right rally", also known as the Charlottesville rally or Charlottesville riots, was a white supremacist rally that occurred in Charlottesville, Virginia, from August 11 to 12, 2017. The rally occurred amidst the backdrop of controversy generated by the removal of Confederate monuments throughout the country in response to the Charleston church shooting in 2015. The event turned violent after protesters clashed with counter-protesters, leaving over 30 injured. The rally also attracted wide attentions on Twitter. Twitter users led vigilante campaigns on the platforms to personally identify and denounce individual marchers in the rally; following the start of the campaign, many of the marchers were shamed and vilified by the social media community, with several of the rally attendees being dismissed from their jobs as a result of the campaign.

Although the rally occurred in Charlottesville originally, messages and/or comments related to it are immediately spread out through Twitter to users in many other places, including all major cities in US, which inspired subsequent vigils and demonstrations in a number of cities across the country in the following days from Aug. 11 and 12, 2017. To this event, we collect a time series of user level information (during the time from Aug. 11 to Sep. 4, 2017) that recorded all Twitter user accounts in 20+ cities that spread, at least once, any message/comment related to the rally during the collection period. We also collect the reaction time of every user to relevant messages and the user-specific information such as the number of followers, friends that an user has and how many tweets the user has published in the past (history posts). In addition, the registration location of the Twitter account and its corresponding latitude and longitude are also collected.

Similar to most rumor spreading data, it is not possible to track how every single message is spread from user to user by our collected data; thus, there is no way to directly identify the interaction network among users. But it is possible to generate the distribution flows of users who have joined the spreading process. Formally, we can define at each time point t that a user has joined the process if and only if by t he/she has at least reacted once to the messages/comments related to the rally; then, the data set can be easily converted to day-by-day distribution flows where, at every time (day) t since the origin (Aug. 11, 2017), we have an N-dimensional $\{0,1\}$-valued vector with N being the number of all users in record. The ith coordinate takes value 1 if and only if the ith user has reacted to the rally-message at least once by t.

For such a distribution flow data set, we are interested in making inference of features of the interaction network between users, because they are useful for making prediction for the other spreading processes on Twitter regarding the similar social events. To that end, we apply the NPSML method to estimate the hidden interaction network from the flow data. Since there are 100,000+ users in our record, and it is likely that many users belong to the same latent group so that their response pattern is similar to their common group members, it is more appropriate to assume the interaction network behind our flow data is a block network and then apply the NPSML to the block network model discussed in Section 3.5.

To uncover the dependence of interaction links between users on their geographical features and/or friendship/followership relation, we embed nodes(users) of the interaction network into a 5-dimensional feature space with the coordinates representing the latitude, longitude of account location, the number of friends, followers and history posts, respectively. To reduce the computation burden, we adopt the bootstrap method, randomly pick 10,000 users from the full set of users for 10 times and estimate the block network on each of the subsamples. For every subsample, an estimator for membership weight function E^1 and interaction matrix E^2 can be derived. The aggregated estimator for interaction matrix E^2 is averaged over all subsample estimators; for the block membership weight E^1, the aggregated estimator is derived by maximum a posteriori from the set of subsample estimators.

For robustness check, we select $dt \in \{0.01, 0.05, 0.1, 0.2\}$ to solve (9). As block network is used, there is no need to draw the M_2 samples of node pairs, only M_1 sampled nodes are needed for evaluating r. To reduce computation burden, we consider to take a much smaller M_1 than the number of all users in record (10,000+) to approximate the membership weight function E^1 and distribution function r. To check the robustness of our estimation with respect to different choice of M_1, we preliminarily run the estimation program on a set of different $M_1 \in \{50, 100, 200, \ldots, 500\}$. The feature vector of the M_1 nodes in each trail is selected by conducting a K-means clustering on the full sample with the number of clusters equal to M_1, then the set of cluster centres will be selected as the feature vector. Such selected feature vector for the M_1 nodes distributes asymptotically in the same way within the feature space as for the full sample of nodes. The preliminary result shows that the estimators are not sensitive to different choice of dt and become stable when M_1 is greater than 50. Therefore, we will fix $dt = 0.2$ and $M_1 = 100$; the 100 cluster centres are also used as the evaluation nodes for the estimated function E^1.

The choice of best block number is still based on maximization of BIC value. We plot the BIC for the three cases that the block number equals to 3, 4, and 5 in Figure 2 and the BIC reaches its maximum when block number is 4, so we consider a block network with 4 blocks as the final model for further analysis.

Different visualizations of the block network are provided. Figure 3 sketches the geographic range of every block/community of the Twitter network; the amount of followers, friends and history posts is plotted along with

Table 2: Mean features of 4 communities.

	Followers	Friends	History posts	Lat	Lon
Big name community	1474739	123835	149494	30.78	-89.99
Famous active community	535641	25967	137372	34.18	-117.59
Famous inactive community	500197	3519	102222	40.75	-82.55
Nobody community	21658	3770	113593	46.77	-122.46

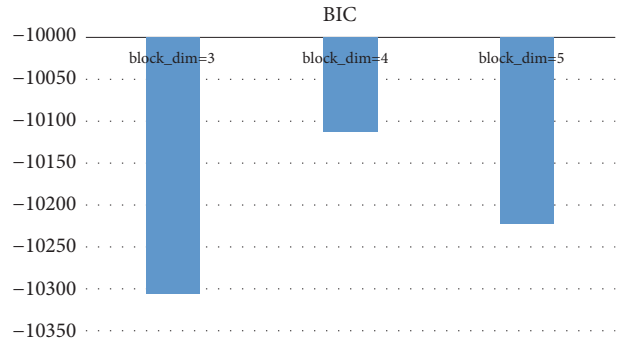


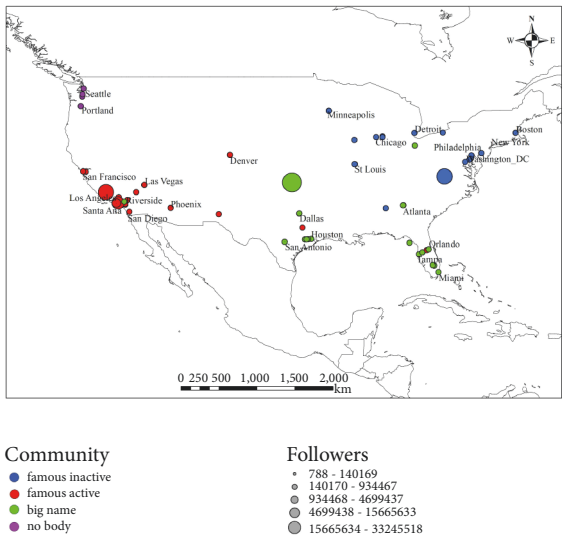
Figure 2: BIC for different block numbers.

locations of every user within every community in subfigures (a), (b) and (c), respectively. Note that the 100 users in plot 3 are synthetic in the sense that their attributes are described by the centre vectors of 100 clusters yielded from applying K-means clustering to the full set of 10,000+ users. Because the clustering is taken on a 5-dimensional feature space, the location of every synthetic user may not lie exactly within a city in the US, nor around a group of neighboring cities. Although the deviation between synthetic users and real users seems to be anomalous, it does reflect the information loss when the higher-dimensional cluster is projected to a low-dimensional space; this part of lost information can play a critical role in determining the community membership of both the synthetic and real users. To see this, consider the synthetic user represented by the largest green dot in Figure 3(a); its geographic location is obviously not close to every city or cities group within our record. To be grouped into the same cluster by K-means method, all real users corresponding to this synthetic user have to have the property that they are quite far away from each other geographically but highly analogous in the other dimension of features, such as the number of followers in this case. Consequently, the community membership of the giant green-dot user and the real users represented by it is not fully determined by geographic factors, while it is more likely to depend on the extra social factors, such as the amount of followers, which are not directly related to users' locations. This observation also justifies the necessity of including extra information into the analysis of information spreading process on Twitter.

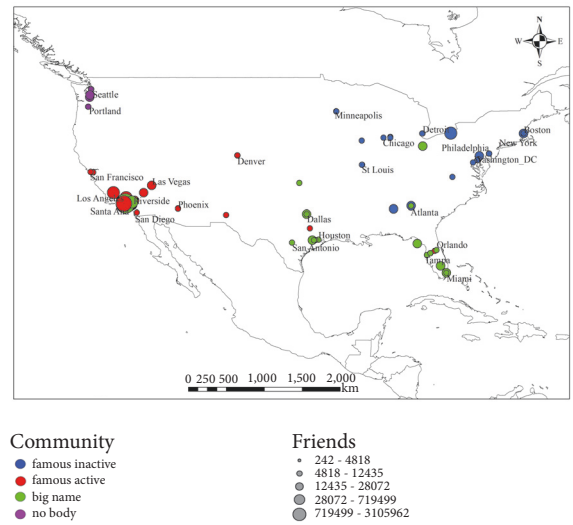
From the mean value of every feature reported in Table 2, the four user communities can be roughly summarized by their activeness as follows: (1) big name community, within which the users are more likely to have a giant group of followers and friends; meanwhile, they are highly active on Twitter; (2) nobody community, within this community users have a fairly small number of followers and friends compared to the other three communities; their history posts are not quite active either; (3) famous inactive community, users in this community have quite a lot of followers, but only a few friends and a relatively small amount of history posts, so this group of users might be "stars" in some fields (large follower group), but they are less likely to interact with the others on Twitter and therefore are not active; (4) famous active community, users in this community do have many followers, but different from inactive community, the average number of friends and history posts is huge, which indicates that they are very active on Twitter.

If we further exam the spatial distribution of features within every community in Figure 3, it is found that (1) for the amount of followers and friends, their spatial distribution is highly uneven within every community, there are only one or two synthetic users with extremely large value, this uneven distribution pattern suggests a classical centre-periphery structure within a community, and the users with greatest amount of followers and/or friends are leaders for the spreading of opinions within their own community and across different communities; (2) the amount of history posts is much more evenly distributed within all the four communities, which reflects the important characteristics of social media that every user on it has the same right to express their own opinion, no matter whether or not they are famous or influential in the real life; (3) although users within every community are not gathered spatially, there exists a weak spatial segregation pattern of the four communities (the segregation can be better visualized in Figure 4); to better understand the source of the spatial segregation, future studies are needed.

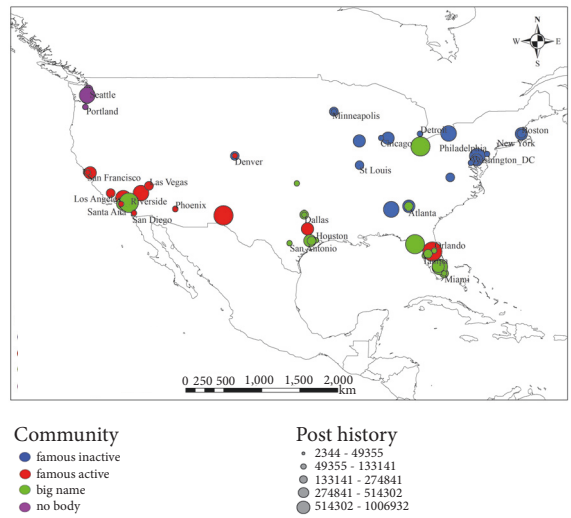
The link strength between different communities is presented in Table 3 (the "From" label in the column header indicates that values in each column representing the impact strength from the community in the column header to the other communities; the "To" label in the row name indicates that values in each row representing the impact strength from the other communities to the community in the row label) and visualized in Figure 4. Apparently, a significant hierarchical structure can be concluded from the link matrix, big name community dominates all the other communities in terms of their sensitivity to social opinions, followed by the famous active community. But compared to the famous active community, the big name community is more likely to accept arguments sourced from the nobody and famous inactive community. For famous inactive community, they only read the tweets posted by members in the big name and famous active communities and receive nothing from its insiders and users from nobody community; this observation



(a) Spatial distribution of followers number within different communities



(b) Spatial distribution of friend numbers within different communities



(c) Spatial distribution of history post within different communities

Figure 3: Spatial distribution of features of users within different communities.

Table 3: Link matrix of 4 communities.

	From big name community	From famous active community	From famous inactive community	From nobody community
To big name community	1	1	1	1
To famous active community	1	1	0.701	0.637
To famous inactive community	0.175	0.365	0	0
To nobody community	0	0	0	0.01

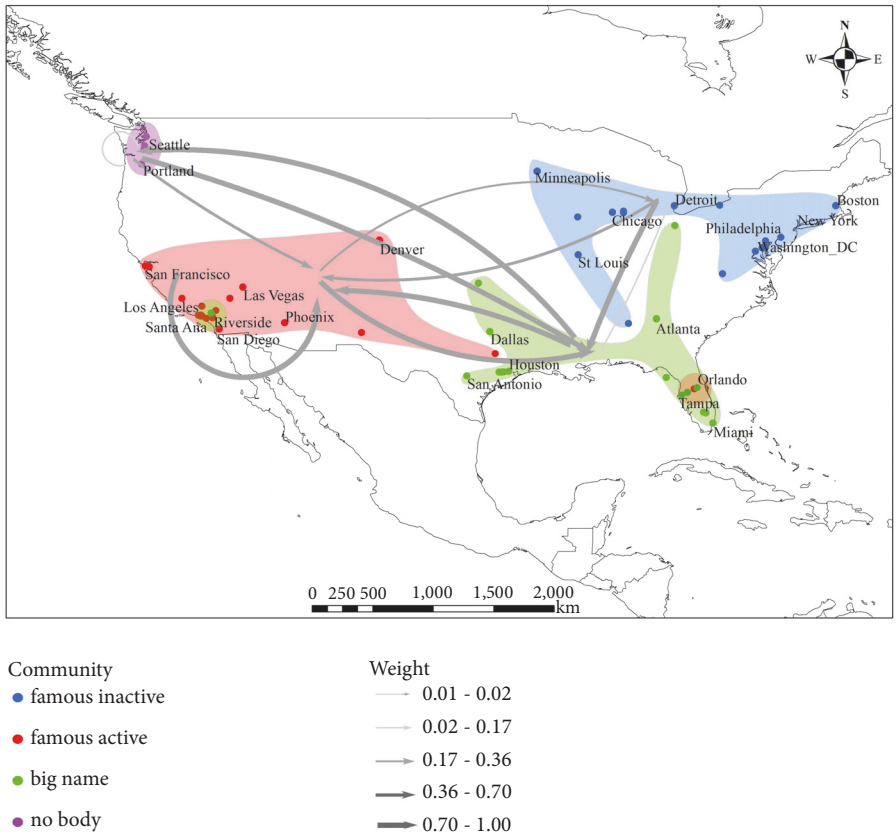


Figure 4: Estimate for interaction matrix.

reflects some kind of opinion discrimination. Finally, the nobody community seems to be isolated from all the other communities and only hear from its insiders, which forms another form of opinion discrimination [54].

From above analysis, there have been quite a few interesting features that can be drawn out of the information spreading process on Twitter. To better understand the formation of the four communities and the hierarchical structure of link matrix, it should be helpful to do more textual mining work on the tweet articles involved in the spreading process and add the extracted information as covariate to the spreading process and reestimate the hidden block network. To do so, a semiparametric extension of the network estimators in this paper is needed; we leave this challenge for future researches.

6. Conclusion and Future Direction

In this paper, we propose a novel approach to nonparametrically estimate the hidden interaction network behind an information spreading process. This approach is designed to handle such an important feature of information spreading processes that the specific spreading trajectory does not exist, and only the distribution flow of the spreading status is observable. To characterize the formation of distribution flows, a mean-field process/equation is proposed. A nonparametric simulation-based maximum likelihood estimator is developed to resolve the subtlety induced by the mean-field equation and the fully nonparametric network edge function. Our estimation procedure can also be applied to the block network structure, a special case of the fully nonparametric network.

To our best knowledge, our work is the first attempt to implement a fully nonparametric estimation of the network structure for distribution flow data and information spreading process. The resulting estimator is always valid if the spreading process is repeatedly observable, while for those spreading processes that are not possible to be repeatedly observed, the estimator turns out still valid in the sense that it is identifiable up to a compact convex set for a fully nonparametric network and completely identifiable for block network under a generic constraint. Therefore for block network, the consistency and asymptotic normality can always be established in the standard way, which is enough for practical use.

Numerical experiments are conducted to verify the effectiveness of our estimation procedure; its practical usefulness is illustrated by a real data application, where the spreading process of tweet articles regarding the event "Unite the Right rally" is studied, and a block network is fitted. The fitting result shows that Twitter users involved in the spreading process can be divided into four communities, which correspond to big name users, famous active and inactive users, and nobody users. Connections among these four communities display a remarkable hierarchical structure; opinion discrimination exists as expected among different communities.

There are some limitations of the current studies: first, we only show that the fast algorithm is efficient in lifting the computation speed when the number of observation times is relatively small compared to the total number of nodes, but a low observation frequency might enlarge the estimation bias. In practice, how to balance the estimation accuracy and the computation is tricky, and further studies are needed. Second, high frequent observation may not always be possible in many applications. In the Twitter data analyzed in this paper, the exact time of posting is available, which makes it possible to extract arbitrarily high frequent distribution flows from the given data. But, in many other applications, the distribution flows are stored in the form of a series of snapshots with fixed length of observational interval. In that case, the observation frequency is strictly controlled by the interval length and not stretchable at all, for which how to develop a reasonable algorithm is still an open question. Third, as mentioned in Section 3.6, the complete identifiability for the fully nonparametric network is not achievable. So constraints are needed to guarantee the desired identifiability. Although, as shown in Remark 2, sparsity is a good constraint to lead identifiability, it may not always be reasonable. Therefore, a further study on the feasible and proper identification condition should be very meaningful in both theoretical and practical aspects.

Data Availability

The data sample and Python code used in this article are available per request from the corresponding author, through xiaoqizh@buff alo.edu.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this manuscript.

Authors' Contributions

Conceptualization was carried out by Xiaoqi Zhang, Yanqiao Zheng, and Xinyue Ye; methodology is done by Xiaoqi Zhang and Xiaobing Zhao; software is contributed by Xiaoqi Zhang; validation is done by Yanqiao Zheng and Xinyue Ye; formal analysis is carried out by Xiaoqi Zhang, Xiaobing Zhao, and Qiwen Dai; investigation is done by Yanqiao Zheng; resources are contributed by Xiaobing Zhao and Xinyue Ye; data curation is done by Xinyue Ye; original draft preparation is carried out by Xiaoqi Zhang and Yanqiao Zheng; review and editing is done by Xinyue Ye and Yanqiao Zheng; visualization is done by Qiwen Dai; supervision is provided by Xiaobing Zhao; project administration is done by Xiaobing Zhao and Xinyue Ye; funding acquisition is carried out by Xiaobing Zhao.

Acknowledgments

This work was partially supported by the China National Planning Office of Philosophy and Social Sciences (18BTJ023). This work was presented at the 15th Xiang'Zhang Economic Forum Seminar (Beijing); the (co-)authors received valuable comments from Dr. Yougui Wang and Zhigang Cao.

References

[1] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang, "TrajGraph: a graph-based visual analytics approach to studying urban network centralities using taxi trajectory data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 160–169, 2016.

[2] C. Yang, M. Xiao, X. Ding et al., "Exploring human mobility patterns using geo-tagged social media data at the group level," *Journal of Spatial Science*, pp. 1–18, 2018.

[3] S. Al-Dohuki, Y. Wu, F. Kamw et al., "SemanticTraj: a new approach to interacting with massive taxi trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 11–20, 2017.

[4] L. Duan, X. Ye, T. Hu, and X. Zhu, "Prediction of suspect location based on spatiotemporal semantics," *ISPRS International Journal of Geo-Information*, vol. 60, no. 7, p. 185, 2017.

[5] S. Han, F. Ren, C. Wu, Y. Chen, Q. Du, and X. Ye, "Using the tensorflow deep neural network to classify mainland china visitor behaviours in hong kong from check-in data," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, p. 158, 2018.

[6] L. Huang, Y. Wen, X. Ye, C. Zhou, F. Zhang, and J. Lee, "Analysis of spatiotemporal trajectories for stops along taxi paths," *Spatial Cognition & Computation*, pp. 1–23, 2018.

[7] X. Shi, B. Xue, M.-H. Tsou et al., "Detecting events from the social media through exemplar-enhanced supervised learning," *International Journal of Digital Earth*, 2018.

[8] Z. Wang and X. Ye, "Space, time and situational awareness in natural hazards: a case study of hurricane sandy with social media data," *Cartography and Geographic Information Science*, 2018.

[9] F. Chierichetti, S. Lattanzi, and A. Panconesi, "Rumor spreading in social networks," *Theoretical Computer Science*, vol. 412, no. 24, pp. 2602–2610, 2011.

[10] N. Song and L. Huo, "Dynamical interplay between the dissemination of scientific knowledge and rumor spreading in emergency," *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 73–84, 2016.

[11] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[12] Z. Chen, *An agent-based model for information diffusion over online social networks [PhD thesis]*, Kent State University, 2016.

[13] J. Lee and X. Ye, "An open source spatiotemporal model for simulating obesity prevalence," in *GeoComputational Analysis and Modeling of Regional Systems*, Advances in Geographic Information Science, pp. 395–410, Springer International Publishing, Cham, Switzerland, 2018.

[14] X. Ye, L. Dang, J. Lee, M. Tsou, and Z. Chen, "Open source social network simulator focusing on spatial meme diffusion," in *Human Dynamics Research in Smart and Connected Communities*, Human Dynamics in Smart Cities, pp. 203–222, Springer International Publishing, Cham, Switzerland, 2018.

[15] W. Luo, D. A. Katz, D. T. Hamilton et al., "Development of an agent-based model to investigate the impact of HIV self-testing

programs on men who have sex with men in atlanta and seattle," *JMIR Public Health and Surveillance*, vol. 4, no. 2, article e58, 2018.

[16] L. Allen, F. Brauer, P. J. Van den Driessche, and J. Wu, *Mathematical Epidemiology*, vol. 1945, Springer, 2008.

[17] L. J. Zhao, J. J. Wang, Y. C. Chen, Q. Wang, J. Cheng, and H. Cui, "SIHR rumor spreading model in social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 7, pp. 2444–2453, 2012.

[18] X. Qiu, L. Zhao, J. Wang, X. Wang, and Q. Wang, "Effects of time-dependent diffusion behaviors on the rumor spreading in social networks," *Physics Letters A*, vol. 380, no. 24, pp. 2054–2063, 2016.

[19] F. Jia and G. Lv, "Dynamic analysis of a stochastic rumor propagation model," *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 613–623, 2018.

[20] M. Cristelli, L. Pietronero, and A. Zaccaria, "Critical overview of agent-based models for economics," https://arxiv.org/abs/1101.1847.

[21] W. Luo, "Visual analytics of geo-social interaction patterns for epidemic control," *International Journal of Health Geographics*, vol. 15, no. 1, article 28, 2016.

[22] W. Luo, P. Gao, and S. Cassels, "A large-scale location-based social network to understanding the impact of human geo-social interaction patterns on vaccination strategies in an urbanized area," *Computers, Environment and Urban Systems*, vol. 72, pp. 78–87, 2018.

[23] K. Ma, W. Li, Q. Guo et al., "Information spreading in complex networks with participation of independent spreaders," *Physica A: Statistical Mechanics and Its Applications*, vol. 492, pp. 21–27, 2018.

[24] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[25] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: a complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.

[26] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[27] B. H. Spitzberg, "Toward a model of meme diffusion (M3D)," *Communication Theory*, vol. 24, no. 3, pp. 311–339, 2014.

[28] W. Härdle, *Applied Nonparametric Regression*, Econometric Society Monographs, no. 19, Cambridge University Press, 1990.

[29] D. Kristensen and Y. Shin, "Estimation of dynamic models with nonparametric simulated maximum likelihood," *Journal of Econometrics*, vol. 167, no. 1, pp. 76–94, 2012.

[30] M. E. J. Newman and E. A. Leicht, "Mixture models and exploratory analysis in networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 104, no. 23, pp. 9564–9569, 2007.

[31] L. Lü and T. Zhou, "Link prediction in complex networks: a survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[32] M. Salter-Townshend, A. White, I. Gollini, and T. B. Murphy, "Review of statistical network analysis: models, algorithms, and software," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 4, pp. 243–264, 2012.

[33] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. Xing, and T. Jaakkola, "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in *Proceedings of the International Biometrics Society Annual Meeting*, vol. 15, 2006.

[34] P. Winker and M. Gilli, "Indirect estimation of the parameters of agent based models of financial markets," FAME Working Paper No. 38, FAME International center for financial asset management and engineering, 2001.

[35] J. Grazzini and M. Richiardi, "Estimation of ergodic agent-based models by simulated minimum distance," *Journal of Economic Dynamics & Control*, vol. 51, pp. 148–165, 2015.

[36] J. Kukacka and J. Barunik, "Estimation of financial agent-based models with simulated maximum likelihood," *Journal of Economic Dynamics & Control*, vol. 85, pp. 21–45, 2017.

[37] T. Zhou, Z. Kuscsik, J. Liu, M. Medo, J. R. Wakeling, and Y. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 107, no. 10, pp. 4511–4515, 2010.

[38] C. Matias, T. Rebafka, and F. Villers, "A semiparametric extension of the stochastic block model for longitudinal networks," *Biometrika*, vol. 105, no. 3, pp. 665–680, 2018.

[39] P. Bickel, D. Choi, X. Chang, and H. Zhang, "Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels," *The Annals of Statistics*, vol. 41, no. 4, pp. 1922–1943, 2013.

[40] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, "Reconstructing propagation networks with natural diversity and identifying hidden sources," *Nature Communications*, vol. 5, article 4323, 2014.

[41] Y. Roudi and J. Hertz, "Mean field theory for nonequilibrium network reconstruction," *Physical Review Letters*, vol. 106, no. 4, 2011.

[42] H. H. M. Weerts, A. G. Dankers, and P. M. J. Van den Hof, "Identifiability in dynamic network identification," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1409–1414, 2015.

[43] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye, "Network reconstruction based on evolutionary-game data via compressive sensing," *Physical Review X*, vol. 1, no. 2, Article ID 021021, pp. 1–7, 2011.

[44] D. Hayden, Y. H. Chang, J. Goncalves, and C. J. Tomlin, "Sparse network identifiability via compressed sensing," *Automatica*, vol. 68, pp. 9–17, 2016.

[45] C. Viboud, O. N. Bjørnstad, D. L. Smith, L. Simonsen, M. A. Miller, and B. T. Grenfell, "Synchrony, waves, and spatial hierachies in the spread of influenza," *Science*, vol. 312, no. 5772, pp. 447–451, 2006.

[46] N. J. Gordon, D. J. Salmond, and S. Adrian, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2, pp. 107–113, 1993.

[47] P. D. Moral, "Measure-valued processes and interacting particle systems. application to nonlinear filtering problems," *The Annals of Applied Probability*, vol. 80, no. 2, pp. 438–495, 1998.

[48] T. Tanaka, "A theory of mean field approximation," in *Advances in Neural Information Processing Systems*, pp. 351–360, 1999.

[49] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[50] P. Del Moral, *Mean Field Simulation for Monte Carlo Integration*, Chapman and Hall/CRC, 2013.

[51] M. A. Golberg, "The derivative of a determinant," *The American Mathematical Monthly*, vol. 79, no. 11, pp. 1124–1126, 1972.

[52] P. K. Andersen, L. S. Hansen, and N. Keiding, "Non-and semi-parametric estimation of transition probabilities from censored observation of a non-homogeneous markov process," *Scandinavian Journal of Statistics*, vol. 18, no. 2, pp. 153–167, 1991.

[53] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[54] J.-V. Cossu, V. Labatut, and N. Dugué, "A review of features for the discrimination of twitter users: application to the prediction of offline influence," *Social Network Analysis and Mining*, vol. 6, no. 1, p. 25, 2016.