Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

WEAVING MULTI-AGENT MODELING AND BIG DATA FOR STOCHASTIC PROCESS INFERENCE

Wen Dong

Department of Computer Science and Engineering &
Institute of Sustainable Transportation and Logistics
State University of New York at Buffalo, USA

ABSTRACT

In this paper, we develop a stochastic process tool to tell the stories behind big data with agent-based models. Specifically, we identify an agent-based model as a stochastic process that generates the big data, and make inferences by solving the agent-based model under the constraint of the data. We hope to use this tool to create a bridge between those who have access to big data and those who use agent-based simulators to convey their insight about these data.

1 INTRODUCTION

"Big data" have the big potential to revolutionize both our understanding of complex systems and our way of making data products. However, it is a significant endeavor to tell stories about big data with generative models (agent-based or system dynamics models) and to implement data products as optimization problems involving big data, because those who work with big data and those who work with models speak different languages.

To bridge big data and generative models, we start by identifying a generative model as a stochastic process, which generates time series according to a well-defined probability measure. We then fit the parameters of the stochastic process to the data using machine-learning methods, and reason about the data by solving the fitted model under the constraints of these data. In stochastic process terminology, the system state of this generative model is comprised of the states of the individual agents, the inter-agent relationships, and the world. System state changes over time according to a state transition matrix/kernel defined by the events of the generative model, and we are interested primarily in making inferences regarding the stochastic process by studying the noisy observations about these stochastic process states.

However, the primary difficulty in solving the stochastic processes defined by an agent-based model—finding the probabilities for the agent-based model to generate every possible state sequences—stems from the exploding state space: in a system with 100 agents, and with each agent taking two states, we will have 2^{100} system states. To avoid the necessity of enumerating state combinations, we introduce mean-field approximation. The estimated probability that I catch a cold is the average probability that I catch a cold from my neighbor x (who is sniffling but whose cold-or-not I don't know) plus the average probability that I catch a cold from my neighbor y (who appears healthy, but whose cold-or-not I also don't know), and so on. The fact that my neighbor x does have a cold will not change my estimation, although it does increase my chance of catching a cold. This kind of mean-field approximation is employed widely in statistical physics, and can be a significant help here in dealing with exponentially-increasing state space (Heskes and Zoeter 2002, Wainwright and Jordan 2008). Other potential methods include Markov chain Monte Carlo (Dong, Heller, and Pentland 2012, Wilkinson 2006) and diffusion approximation (Wilkinson 2006).

This paper therefore advocates, for the first time, that we should combine the power of big data and the power of model-thinking in the stochastic process framework. Agent-based modeling is a physicist's approach for modeling human societies when data are unavailable and experiments impossible (Epstein

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

2007), and we believe that big data will transform agent-based modeling from speculation into a physical science. This paper also offers a solution to fit any agent-based model defined by a production rule system to big time series data, based on mean-field approximation. Hence, this system bridges modelers with data miners. We have benchmarked our solution on systems of hundreds of agents, and our benchmarking gives meaningful results.

With the tool to weave multi-agent modeling and big data for stochastic process inference, we hope to track real world social systems continuously using big data from social media and Internet of Things on the one hand, and multi-agent models on the other hand. We can better understand the diffusion of influence and the formation of networks (Dong, Lepri, and Pentland 2011, Dong, Heller, and Pentland 2012) from the behavior and interaction data continuously observed by personal mobile phones (MIT Human Dynamics Lab 2012), and social physics models (Castellano, Fortunato, and Loreto 2009). We can estimate road traffic (Dong and Pentland 2009) in real-time according to millions of tracked vehicle locations and multi-agent transportation simulators ((Smith, Beckman, and Baggerly 1995, MATSim development team (ed.) 2007)). We can track national socio-economical indicators according to call detail records (Orange S. A.) and city models (Batty 2007).

The rest of this paper is organized as follows. In Section 2, we introduce a probabilistic production (rule) system to describe the microscopic dynamics of a generative model, and identify the production system as a stochastic process. In Section 3, we derive a mean-field solution to the generative model under the constraint of data—in other words, given a simulator and noisy observational data about a process generated by the simulator logic, our algorithm infers the probabilities on a per-agent basis of all possible outcomes. In Section 4, we give examples and benchmark this algorithm against other algorithms. We summarize what we have accomplished and offer our speculation about big data in Section 5.

2 STOCHASTIC PROCESS INDUCED BY PRODUCTION SYSTEM

We employ a production (rule) system to describe multi-agent dynamics. Such a production system is widely used in artificial intelligence, chemistry, systems biology, and other fields.

A production system in artificial intelligence consists of a working memory that maintains the current state of the world and a rule interpreter that triggers different rules (or productions) with a well-defined mechanism, changing the state of the world according to these triggered rules. A production has a precondition and an action—when the precondition matches the current state of the world, the interpreter can trigger the production, execute the action of the production, and so change the current state of the world. Researchers in expert systems are interested in using a determinist algorithm that triggers the best rule for reaching a goal from a collection of matching rules. In comparison, we treat a production system as a stochastic process by assigning different rates of happening to rules, and by defining a time series of system states as the probability of a sequence of rules firing, which generates the system-state time-series. We take this approach because we are most interested in fitting the production system to observations about how people behave and interact, in order to solve problems and solve for the rates. This allows us to make predictions about human behavior and interactions using the fitted production system, an effort similar to those in research on cognition architecture (Anderson 1996, Newell and Simon 1972).

Algorithm: Markov process induced by a multi-agent model

Input: initial world state $x\left(t=0\right)$, productions $1,\dots,v$, each production happens with rate $h_k\left(x_t,c_k\right)=c_k g\left(x_t\right)$ and change world state $x_{t^-}\xrightarrow{k}x_t$.

Output: a series of times when productions are triggered, the IDs of the triggered productions, and the corresponding states brought about by the triggered productions $\{t_i,v_i,x\left(t_i\right):i\}$ where $0=t_0<t_1<\cdots<t_n<t_{n+1}=T$, $v_i\in\{1,\dots,v\}$, $x\left(t_i^-\right)\xrightarrow{v_i}x\left(t_i\right)$ and $x\left(t\right)$ is the right limit and the time series $x\left(t\right)$ is left continuous.

Procedure:

Basis set current time to $t_0 = 0$, set the current state to $x(t_0)$, repeat the following step until the current time $t_{i+1} > T$.

Induction sample the next reaction time $\tau \sim$ exponential $\left(\sum_k h_k(x(t_i), c_k)\right)$, sample the next reaction $v_{i+1} \sim h_k / \sum_k h_k$, set current time to $t_{i+1} = t_i + \tau$, and update world state $x(t_{i+1})$ according to production v_{i+1}.

Such a model defines a stochastic process. The probability for this sequence of events $\{t_i, v_i, x(t_i) : i\}$ to happen is $P(v, x) = \prod_i h_{v_i}(x_t) \cdot \exp\left(-\sum_i h_{v_i}(x_t) \cdot (t_{i+1} - t_i)\right)$ (Wilkinson 2006). The time interval τ to the next event is the minimum of exponential distributions, and so is itself an exponential distribution. We leave abstract the actions of the productions, because an inference algorithm with general productions will be more useful.

For example, in the Lotka-Volterra model (commonly known as the predator-prey model), the world is comprised of $x^{(1)}$ number of predators and $x^{(2)}$ number of prey. The prey has an unlimited food supply, and will reproduce with a rate proportional to the prey population without predation (prey $\rightarrow 2 \times$ prey, or $x^{(1)} \rightarrow x^{(1)} + 1$ with rate $c_1 x^{(1)}$), but predation will decrease the prey population and increase the predator population at a rate proportional to the rate that predator and prey meet (prey $+$ predator $\rightarrow 2 \times$ predator, or $(x^{(1)}, x^{(2)}) \rightarrow (x^{(1)} - 1, x^{(2)} + 1)$). The predator population will lose its members due to natural death at a rate proportional to the predator population (predator $\rightarrow \emptyset$, or $x^{(2)} \rightarrow x^{(2)} - 1$). By "solving" the Lotka-Voterra equations—that is, finding the temporal evolution of the predator population and the prey population, constrained by initial and boundary conditions—we can answer questions such as whether the two species could coexist, whether the populations could pass through certain thresholds, and what the likely population sizes will be in the future.

Theoretically, any computer program that operates computer storage through a set of instructions can describe a production system with its world state and rules. However, we prefer the simplest model that can explain the phenomenon under inspection, because the simplest model with the fewest rules may be the most generalized and applicable model for leveraging our past experiences to gain an advantage in the future (Popper 1965). In fact, many elegant models find applications across domains. Epidemics models, for example, are applicable not only to the study of epidemics themselves, but also to the spreading of opinions and innovations (Castellano, Fortunato, and Loreto 2009). Similarly, Lotka-Voterra models are applicable to ecology, but also to economics when researchers want to understand the interactions of different industry sectors (Goodwin 1975).

The stochastic process inference tool developed in this paper aims to create a bridge between big data and business insight about big data, enabling data miners to tell stories behind data with the logic of agent-based models, and enabling agent-based modelers to make predictions about real-world systems from data.

3 MAKING INFERENCES WITH PRODUCTION SYSTEM

In this section, we derive a computationally-tractable solution for any network dynamics model identified by a production system. In other words, given a network dynamics model with unknown event constants $\{c_k : k = 1, \ldots, v\}$ and latent states $\{x_t = (x_t^{(1)}, \cdots, x_t^{(C)}) : t \in R\}$, we find the most likely estimate of the event constants and latent states from a finite number of observations about the model $\{y_t^{(c)} : (c, t)\}$, where the superscript c indicates different agents. We first discretize the Markov jump process into a hidden Markov process in order to make the problem suitable for numerical evaluation and implementation on digital computers. We then apply a mean-field approximation such that the latent state of a node evolves according to the average effects of the latent states of other nodes'. With this approximation in place, we give the expectation maximization (EM) algorithm of the resulting hidden Markov model as a solution to our original problem.

3.1 Time and State Discretization

We can assume that the latent state x_t defined by a production system takes a finite number of values in many applications, and treat a production system as a continuous time Markov chain with transition rate matrix (also called infinitesimal generator) $Q(i,j) = \frac{d}{d\tau}p(x_{t+\tau} = j|x_t = i)$ given by

$$Q(x_t, x_{t+\tau}) = \begin{cases} \sum_{x_t \xrightarrow{k} x_{t+\tau}} h_k(x_t, c_k) & \text{if } x_t \neq x_{t+\tau} \\ 1 - \sum_k h_k(x_t, c_k) & \text{if } x_t \equiv x_{t+\tau} \end{cases}. \tag{1}$$

These latent states may have already taken a finite number of values (e.g., infected vs. susceptible in the SIS epidemics model). When the latent states take only a few integer values "most of time" (i.e., the tightness assumption (Dudley 2002)), we set the domain of the latent states to be these integer values according to the computational precision requirement. When the latent states take an interval of real values most of time, we set the domain to be a few disjoint subintervals, again according to the precision requirement.

A continuous-time Markov chain is related to a discrete-time Markov chain through a uniform rate parameter $\gamma \geq \max_x \sum_k h_k(x, c_k)$ using either the uniformization method (also known as Jensen's method), or the randomization method (Ibe 2008). Let $Q(x_t \xrightarrow{k} x_t + \Delta x_k) = h_k(x_t, c_k)$ be the infinitesimal generator of the continuous-time Markov chain. The probability transition matrix of the uniformized discrete-time Markov chain is $I + Q/\gamma$, where I is the identity matrix and the time of the transitions is a Poisson process with rate γ. The number of transitions between time 0 and time Δt is a random variable of Poisson distribution with rate constant γ—that is, the probability of n transitions during Δt, including null transitions $x \to x$, is $e^{-\gamma\Delta t}(\gamma\Delta t)^n/n!$. The state transition matrix between time 0 and time t is therefore $p(x_0 \to x_{\Delta t}) = \sum_{n=0}^{\infty}(I + Q/\gamma)^n e^{-\gamma\Delta t}(\gamma\Delta t)^n/n!$. When we let $\gamma \to \infty$ and $\gamma\Delta t = 1$, we get $p(x_0 \to x_{\Delta t}) \to I + \frac{Q}{\cancel{\gamma}} \cdot \cancel{\gamma}\Delta t$.

The likelihood of any sample $\{x_t, y_t : t = n\Delta t\}$ generated by a given network dynamics model is

$$p(\{x_t, y_t : t = n\Delta t\}|\{c_k : k = 1, \ldots, v\}) \tag{2}$$

$$= \overset{\text{observation}}{\prod_{n,c} p(y^{(c)}_{n\Delta t}|x^{(c)}_{n\Delta t})} \cdot \overset{\text{events}}{\prod_{n,\Delta x} (\sum_{\Delta x_k = \Delta x} \Delta t \cdot h_k(x_{n\Delta t}, c_k))^{1_{x_{(n+1)\Delta t} - x_{n\Delta t} \equiv \Delta x}}} \cdot \overset{\text{no event}}{\prod_n (1 - \Delta t \cdot \sum_k h_k(x_{n\Delta t}, c_k))^{1_{x_{(n+1)\Delta t} \equiv x_{n\Delta t}}}}.$$

In the following, we will use $t \in \mathbb{N}$ to represent discrete time and $h_k(x_t, c_k)$ to represent the probability for event k to happen in discrete-time hidden Markov model.

3.2 Forward-backward Algorithm under Mean-field Approximation

Recall that a hidden Markov model assigns to every sequence of hidden states $\{x_t : t\}$ and the corresponding observations $\{y_t : t\}$ a probability $p(\{x_t, y_t : t = 1, \ldots, T\})$, according to the state transition probability $p(x_t|x_{t-1})$ and the probabilistic observation distribution $p(y_t|x_t)$. Inferring the probability distribution of latent states involves dynamic programming to iteratively solve the forward statistics $\tilde{\alpha}_t(x_t) = p(x_t, y_{1,\ldots,t})$ for $t = 1, \ldots, T$, the backward statistics $\tilde{\beta}_t(x_t) = p(y_{t+1,\ldots,T})$ for $t = T, \ldots, 1$, and the one-step statistics $\gamma_t(x_t) = p(x_t|y_{1,\ldots,T})$ and two-step statistics $\xi_t(x_{t,t+1}) = p(x_{t,t+1}|y_{1,\ldots,T})$, and from these statistics we can estimate the state transition probability and the parameters of the observation model. To avoid floating-point underflow, we often compute

$$\alpha_t(x_t) = p(x_t|y_{1,\ldots,t}, \pi) = \tilde{\alpha}_t(x_t) \Big/ \prod_{\tau=1}^{t} Z_\tau, \; \beta_t(x_t) = \tilde{\beta}_t(x_t) \Big/ \prod_{\tau=t+1}^{T} Z_\tau, \text{ where}$$

$$Z_t = p(y_t|y_{1,\ldots,t-1}) = \sum_{x_{t-1,t}} \alpha_{t-1}(x_{t-1}) p(x_t|x_{t-1}) p(y_t|x_t) = \sum_{x_{t-1,t}} \alpha_{t-1}(x_{t-1}) p(x_t|x_{t-1}) p(y_t|x_t) \beta_t(x_t),$$

In a hidden Markov model identified by a (stochastic) multi-agent model or a system dynamics model, the state space can be huge, representing the joint state of thousands to millions of agents, or thousands of system variables. In addition, the state-transition probability is normally specified with a limited number of events. In the following, we solve the marginal statistics $\alpha_t^{(c)}(x_t^{(c)})$, $\beta_t^{(c)}(x_t^{(c)})$, $\gamma_t^{(c)}(x_t^{(c)})$ and $\xi_t^{(c)}(x_t^{(c)}, x_{t+1}^{(c)})$ with mean-field approximation — the state of an individual node evolves according to the average (mean-field) effects of the other nodes — and estimate the event rates according to the marginal statistics.

We adopt the approximation

$$\gamma_t^{(c)}(x_t^{(c)}) = \alpha_t^{(c)}(x_t^{(c)}) \beta_t^{(c)}(x_t^{(c)}), \tag{3}$$

$$p(x_t|y_{1,\ldots,T}) = \prod_c \gamma_t^{(c)}(x_t^{(c)}), \tag{4}$$

seek the marginal forward statistics $\alpha_t^{(c)}$ to be consistent with the two-slice statistics $\xi_t(x_{t-1,t})$ for $t = 2, \ldots, T$, and seek the backward statistics $\beta_{t-1}^{(c)}$ to be consistent with the two-slice statistics $\xi_t(x_{t-1,t})$ for $t = T, \ldots, 2$:

$$\begin{aligned}
\text{find } \alpha_t^{(c)} \text{ s.t. } & \sum_{\text{fix } x_t^{(c)}} \xi_t(x_{t-1,t}) = \alpha_t^{(c)}(x_t^{(c)}) \beta_t^{(c)}(x_t^{(c)}), \\
\text{find } \beta_{t-1}^{(c)} \text{ s.t. } & \sum_{\text{fix } x_{t-1}^{(c)}} \xi_t(x_{t-1,t}) = \alpha_t^{(c)}(x_{t-1}^{(c)}) \beta_{t-1}^{(c)}(x_{t-1}^{(c)}), \\
\text{where} \xi_t(x_{t-1,t}) = & \frac{1}{Z_t} \prod_c \alpha_{t-1}^{(c)}(x_{t-1}^{(c)}) \prod_c \beta_t^{(c)}(x_t^{(c)}) p(x_t|x_{t-1}) p(y_t|x_t).
\end{aligned}$$

We further let event rates factorizable into a product of functions, each of which involves one node in the system

$$p(x_t|x_{t-1}) = \begin{cases} \sum_{k: x_t = x_{t-1} + \Delta x_k} c_k \prod_c g_k^{(c)}(x_{t-1}^{(c)}) & x_t = x_{t-1} + \Delta x_k \\ 1 - \sum_k c_k \prod_c g_k^{(c)}(x_{t-1}^{(c)}) & x_t = x_{t-1} \end{cases} \tag{5}$$

For example, the infection rate in the SIS epidemic model $c \cdot x^{(1)} \cdot x^{(2)}$ is the rate when one infected person is in contact with one susceptible person (c) multiplied by the number of different ways in which $x^{(1)}$ susceptible person can be in contact with $x^{(2)}$ infectious persons. Similarly, the predation rate in the predator-prey model $c \cdot x^{(1)} \cdot x^{(2)}$ is the rate at which one predator is in contact with one prey multiplied by the number of different ways that $x^{(1)}$ prey can be in contact with $x^{(2)}$ predators.

We find the marginal two-slice statistics $\xi_t^{(c)}(x_{t-1,t}^{(c)})$ by marginalizing $\xi_t(x_{t-1,t})$ over all $x_{t-1,t}^{(c')}$ for $c' \neq c$:

$$\xi_t^{(c)}(x_{t-1,t}^{(c)}) = \sum_{c' \neq c} \sum_{x_{t-1,t}^{(c')}} \xi_t(x_{t-1,t}) = \frac{1}{Z_t} \cdot \tag{6}$$

$$\Bigg(\overset{\text{probability of no state change}}{(1 - \sum_k c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')} \Big|_{x_{t-1}^{(c')} = x_t^{(c')}})} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')} = x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \alpha_{t-1}^{(c')}(x_{t-1}^{(c')}) \beta_t^{(c')}(x_t^{(c')}) p(y_t^{(c')}|x_t^{(c')}) +$$

$$\sum_k \overset{\text{state transition probability due to event } k}{c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}}} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \alpha_{t-1}^{(c')}(x_{t-1}^{(c')}) \beta_t^{(c')}(x_t^{(c')}) p(y_t^{(c')}|x_t^{(c')}) \Bigg)$$

where

$$g_k^{(c')}\Big|_{\text{constraint}} = \frac{\sum_{\text{constraint}} \alpha_{t-1}^{(c')} g_k^{(c')} \beta_t^{(c')} p(y_t^{(c')}|x_t^{(c')})}{\sum_{\text{constraint}} \alpha_{t-1}^{(c')} \beta_t^{(c')} p(y_t^{(c')}|x_t^{(c')})},$$

$$Z_t = (1 - \sum_k c_k \prod_{c'} g_k^{(c')}\Big|_{x_{t-1}^{(c')}=x_t^{(c')}}) \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}=x_t^{(c')}} \alpha_{t-1}^{(c')}(x_{t-1}^{(c')}) \beta_t^{(c')}(x_t^{(c')}) p(y_t^{(c')}|x_t^{(c')}) +$$

$$\sum_k c_k \prod_{c'} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}} \alpha_{t-1}^{(c')}(x_{t-1}^{(c')}) \beta_t^{(c')}(x_t^{(c')}) p(y_t^{(c')}|x_t^{(c')}).$$

Hence we can make inferences about agent states in a multi-agent model by iteratively solving the marginal forward and backward statistics (Equation 7 and Equation 8), and we can estimate event rate constants from the marginal statistics (Equation 9). The probability of observations under the approximate distribution is $p = \prod_t Z_t$.

$$\alpha_t^{(c)}(x_t^{(c)}) \leftarrow 1/Z_t \cdot \tag{7}$$

$$\Bigg(\overset{\text{probability of no state change}}{(1 - \sum_k c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}=x_t^{(c')}})} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}=x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \alpha_{t-1}^{(c')} \left(\beta_t^{(c')}\right)^{1_{c' \neq c}} p(y_t^{(c')}|x_t^{(c')}) +$$

$$\sum_k \overset{\text{state transition probability due to event } k}{c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}}} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \alpha_{t-1}^{(c')} \left(\beta_t^{(c')}\right)^{1_{c' \neq c}} p(y_t^{(c')}|x_t^{(c')}) \Bigg),$$

$$\beta_{t-1}^{(c)}(x_{t-1}^{(c)}) \leftarrow 1/Z_t \cdot \tag{8}$$

$$\Bigg(\overset{\text{probability of no state change}}{(1 - \sum_k c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}=x_t^{(c')}})} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}=x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \left(\alpha_{t-1}^{(c')}\right)^{1_{c' \neq c}} \beta_t^{(c')} p(y_t^{(c')}|x_t^{(c')}) +$$

$$\sum_k \overset{\text{state transition probability due to event } k}{c_k g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}}} \cdot \prod_{c'} \sum_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}, \text{fix } x_{t-1}^{(c)}} \left(\alpha_{t-1}^{(c')}\right)^{1_{c' \neq c}} \beta_t^{(c')} p(y_t^{(c')}|x_t^{(c')}) \Bigg),$$

$$c_k \leftarrow \frac{1}{C} \cdot \sum_c \sum_t \xi_t^{(c)}(\text{event } k) \Big/ \sum_t \prod_c g_k^{(c)}\Big|_{x_{t-1}^{(c)}=x_t^{(c)}} \tag{9}$$

$$\text{where } \xi_t^{(c)}(\text{event } k) = \xi_t^{(c)}(x_t^{(c)} - x_{t-1}^{(c)} = \Delta x_k^{(c)}) \cdot \frac{c_k^{\text{old}} g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}}}{\sum_{k': \Delta x_{k'}^{(c)}=\Delta x_k^{(c)}} c_k^{\text{old}} g_{k'}^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_{k'}^{(c')}=x_t^{(c')}}}.$$

3.3 Parameter Estimation Under Mean-Field Approximation

In order to determine an estimation for the state transition model (i.e., c_k) and the observation model ($p(y^{(c)}|x^{(c)})$) that maximizes the expected log likelihood in the M-step of the EM algorithm, we set the derivatives of the expected log likelihood over the parameters to be zero and consider constraints when necessary:

$$\frac{\partial \mathbb{E}\log p}{\partial c_k} = \frac{1}{C}\sum_c \sum_t \left(\frac{\xi_t^{(c)}(\text{event } k)}{c_k} - \prod_c g_k^{(c)}\Big|_{x_{t-1}^{(c)}=x_t^{(c)}} \right) \overset{\text{set}}{=} 0,$$

$$\text{where } \xi_t^{(c)}(\text{event } k) = \xi_t^{(c)}(x_t^{(c)} - x_{t-1}^{(c)} = \Delta x_k^{(c)}) \cdot \frac{c_k^{\text{old}} g_k^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_k^{(c')}=x_t^{(c')}}}{\sum_{k':\Delta x_{k'}^{(c)}=\Delta x_k^{(c)}} c_k^{\text{old}} g_{k'}^{(c)}(x_{t-1}^{(c)}) \prod_{c' \neq c} g_k^{(c')}\Big|_{x_{t-1}^{(c')}+\Delta x_{k'}^{(c')}=x_t^{(c')}}}$$

and c_k^{old} is the old estimation of rate constant.

It follows that the new estimation of the rate constant (c_k) of transition k is the total number of transitions in the sample divided by the total number of times the transition could happen:

$$c_k \leftarrow \frac{1}{C} \cdot \sum_c \sum_t \xi_t^{(c)}(\text{event } k) \Big/ \sum_t \prod_c g_k^{(c)}\Big|_{x_{t-1}^{(c)}=x_t^{(c)}} .$$

The maximum likelihood estimation of the parameters of the output models ($p(y^{(c)}|x^{(c)})$) is model-dependent. Here, we provide the solutions for the two most common output models.

When the observation about a node takes a finite number of values, we get

$$p(y^{(c)}|x^{(c)}) = \frac{1_{y_t^{(c)} \equiv y^{(c)}} \sum_t \gamma_t^{(c)}(x^{(c)})}{\sum_t \gamma_t^{(c)}(x^{(c)})} \tag{10}$$

When the observation of a node takes a normal distribution around the state of the node, we have

$$\mu^{(c)}(x^{(c)}) = \sum_t y_t^{(c)} \cdot \gamma_t^{(c)}(x^{(c)}) \Big/ \sum_t \gamma_t^{(c)}(x^{(c)}), \tag{11}$$

$$\sigma^{(c)}(x^{(c)}) = \sum_t (y_t^{(c)} - \mu^{(c)})^2 \cdot \gamma_t^{(c)}(x^{(c)}) \Big/ \sum_t \gamma_t^{(c)}(x^{(c)}). \tag{12}$$

4 EXAMPLES

In this section, we make stochastic process inferences using the predator-prey systems dynamics and the ant forage multi-agent dynamics. The predator-prey systems dynamics are simple enough, such that we can touch on the details of identifying an agent-based model as a stochastic process and make inferences thereof, and of how mean-field approximation works. As such, ant forage multi-agent dynamics nicely demonstrates how mean-field approximation enables us to make inferences in big systems.

4.1 Inferring Predator-Prey Dynamics

The predator-prey (Lotka-Volterra) model taken from (Wilkinson 2006) has three reactions, in which predation increases predator population by one and decrease prey population by one at the same time. "Prey" and "predator" are respectively a prey individual and a predator individual:

$$\begin{aligned} \text{prey} &\rightarrow 2\ \text{prey}, && \text{rate} = c_1 \\ \text{prey} + \text{predator} &\rightarrow 2\ \text{predator}, && \text{rate} = c_2 \\ \text{predator} &\rightarrow \emptyset, && \text{rate} = c_3 \end{aligned}$$

Interpreting this Lotka-Volterra systems dynamics as a discrete-time Markov chain (Section 3.1), the time-dependent system-state is the predator and prey populations $(x_t^{(1)}, x_t^{(2)}) \in \{1, \ldots, M_1\} \times \{1, \ldots, M_2\}$, and the state transition kernel (Equation 13) is induced by 3 reactions, and an event rate can be estimated as the total event occurrence over the number of times the event could have occurred (Equation 14).

In comparison, under mean-field approximation (Section 3.2), we keep track of only the marginal state distributions of the predator and prey populations (Equation 15 and Equation 16), where the probability that the prey population decreases due to predation is computed according to the average predator population $\overline{x_n^{(2)}} = \mathbb{E}_{p(x_t^{(2)})} x_t^{(2)}$, and the probability of predator increase due to predation is computed according to the average prey population $\overline{x_n^{(1)}} = \mathbb{E}_{p(x_t^{(1)})} x_t^{(1)}$. The an event rate (Section 3.3) is estimated as the total event occurrences in both populations over the total number of times the event could have occurred in both populations (Equation 17).

$$p(x_n \rightarrow x_{n+1}) = \tag{13}$$
$$\begin{cases} c_1 x_n^{(1)} & x_{n+1} - x_n = (1, 0) \\ c_2 x_n^{(1)} x_n^{(2)} & x_{n+1} - x_n = (-1, 1) \\ c_3 x_n^{(2)} & x_{n+1} - x_n = (0, -1) \\ 1 - c_1 x_n^{(1)} - c_2 x_n^{(1)} x_n^{(2)} - c_3 x_n^{(2)} & x_{n+1} - x_n = (0, 0) \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} c_1 &= \frac{\sum_{n, x_n} \xi\left(x_n, (x_n^{(1)} + 1, x_n^{(2)})\right)}{\sum_{n, x_n} x_n^{(1)} \xi(x_n, x_n)} \\ c_2 &= \frac{\sum_{n, x_n} \xi\left(x_n, (x_n^{(1)} - 1, x_n^{(2)} + 1)\right)}{\sum_{n, x_n} x_n^{(1)} x_n^{(2)} \xi(x_n, x_n)} \\ c_3 &= \frac{\sum_{n, x_n} \xi\left(x_n, (x_n^{(1)}, x_n^{(2)} - 1)\right)}{\sum_{n, x_n} x_n^{(2)} \xi(x_n, x_n)}. \end{aligned} \tag{14}$$

$$p(x_n^{(1)} \rightarrow x_{n+1}^{(1)}) = \tag{15}$$
$$\begin{cases} c_1 x_n^{(1)} & x_{n+1}^{(1)} = x_n^{(1)} + 1 \\ c_2 x_n^{(1)} \overline{x_n^{(2)}} & x_{n+1}^{(1)} = x_n^{(1)} - 1 \\ 1 - c_1 x_n^{(1)} - c_2 x_n^{(1)} \overline{x_n^{(2)}} & x_{n+1}^{(1)} = x_n^{(1)} \\ 0 & \text{otherwise} \end{cases}$$

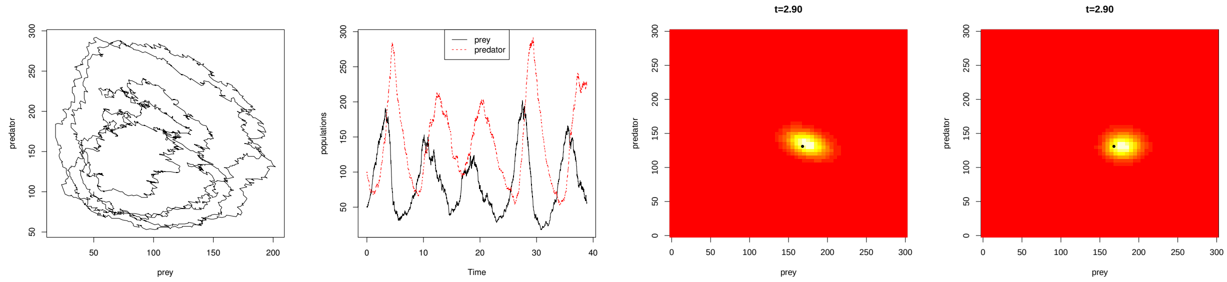
$$p(x_n^{(2)} \rightarrow x_{n+1}^{(2)}) = \tag{16}$$
$$\begin{cases} c_3 x_n^{(2)} & x_{n+1}^{(2)} = x_n^{(2)} - 1 \\ c_2 \overline{x_n^{(1)}} x_n^{(2)} & x_{n+1}^{(2)} = x_n^{(2)} + 1 \\ 1 - c_2 \overline{x_n^{(1)}} x_n^{(2)} - c_3 x_n^{(2)} & x_{n+1}^{(2)} = x_n^{(2)} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} c_1 &= \frac{\sum_{n, x_n^{(1)}} \xi^{(1)}(x_n^{(1)}, x_n^{(1)} + 1)}{\sum_{n, x_n^{(1)}} x_n^{(1)} \xi^{(1)}(x_n^{(1)}, x_n^{(1)})} \\ c_2 &= \frac{\sum\limits_{n, x_n^{(1)}} \xi^{(1)}(x_n^{(1)}, x_n^{(1)} - 1) + \sum\limits_{n, x_n^{(2)}} \xi^{(2)}(x_n^{(2)}, x_n^{(2)} + 1)}{\sum\limits_{n, x_n^{(1)}} x_n^{(1)} \xi^{(1)}(x_n^{(1)}, x_n^{(1)}) \overline{x_n^{(2)}} + \sum\limits_{n, x_n^{(2)}} x_n^{(2)} \xi^{(2)}(x_n^{(2)}, x_n^{(2)}) \overline{x_n^{(1)}}} \\ c_3 &= \frac{\sum_{n, x_n^{(2)}} \xi^{(2)}(x_n^{(2)}, x_n^{(2)} - 1)}{\sum_{n, x_n^{(2)}} x_n^{(2)} \xi^{(2)}(x_n^{(2)}, x_n^{(2)})}. \end{aligned} \tag{17}$$

With the discrete-time Markov chains of both the predator-prey model and the approximate model, we can proceed to compare the inferences that use mean-field approximation with the exact inferences. Mean-field approximation is one way to pay the price of being able to make stochastic process inferences on dynamical systems involving an intractable number of interacting factors or agents.

We set $c_1 = 1$, $c_2 = .007$, $c_3 = .6$, $x_0^{(1)} = 50$, $x_0^{(2)} = 100$, and sampled a time series of prey/predator populations from time 0 to time 40 according to a continuous-time Markov process (Section 2). The time series show periodicity, randomness, and predator population following prey population (Figure 1a, 1b). We perturbed the system states in time intervals $[2, 2.1)$ and $(7.7, 8]$ with a Gaussian random noise of mean 0 and a standard deviation of 20 ($\mathcal{N}(0, 20^2)$) (resulting in noisy observations), and compared the exact inference population trajectories between time 2.1 and time 7.7 (Figure 1c) with the mean-field inference (Figure 1d).

Within 0.5 time unit of observations (i.e., at times 2.5 and 7.5), the inferred predator and prey populations using exact forward-back inference are visibly correlated. This correlation results from predation decreasing prey population and increasing predator population at the same time. The correlation between populations becomes less visible as we move away in time from the observations. The mean-field approximation removes the correlation by decoupling the predation equation into two: decreasing the prey population due to the existence of the predator population, and increasing predator due to the existence of prey.



(a) phase space

(b) time series

(c) exact inference

(d) mean-field inference

Figure 1: Predator-prey dynamics (a) in phase space, (b) as time series, (c) population density estimation (c) from exact inference, and (d) mean-field inference.

4.2 Inferring Ant Forage Dynamics

In this section, we make stochastic process inferences using ant forage multi-agent dynamics. The ant forage dynamics under discussion is from The NetLogo model "Ants," defined by NetLogo programming language (Wilensky 1997). The Ants world is a 71 × 71 grid with an ant nest at the center, three food sources (each with different amounts of food and at different distances from the nest), and 125 ants moving food from the sources to the nest. The logic of individual ants is simple, but it nonetheless adds up to an interesting collective intelligence. An ant with food wiggles toward the nest and leaves a specified amount of chemical on grid points along its way. This chemical diffuses to the eight neighboring grids and then evaporates. An ant without food senses the chemical with moderate density in front of it, and follows the chemical to an area with high chemical concentration. In areas with too low or too high chemical concentrations, the ant merely wiggles forward until it finds food. A simulation run of the Ants model is animated in Figure 2a.

Agent-based modelers simulate the Ants model without engaging any observational data. Here, we show that we can translate the ant forage model in NetLogo into a discrete-time Markov chain, and infer how ants travel from their origins at time 0 to their destinations at time 200, considering their interactions through chemical and food.

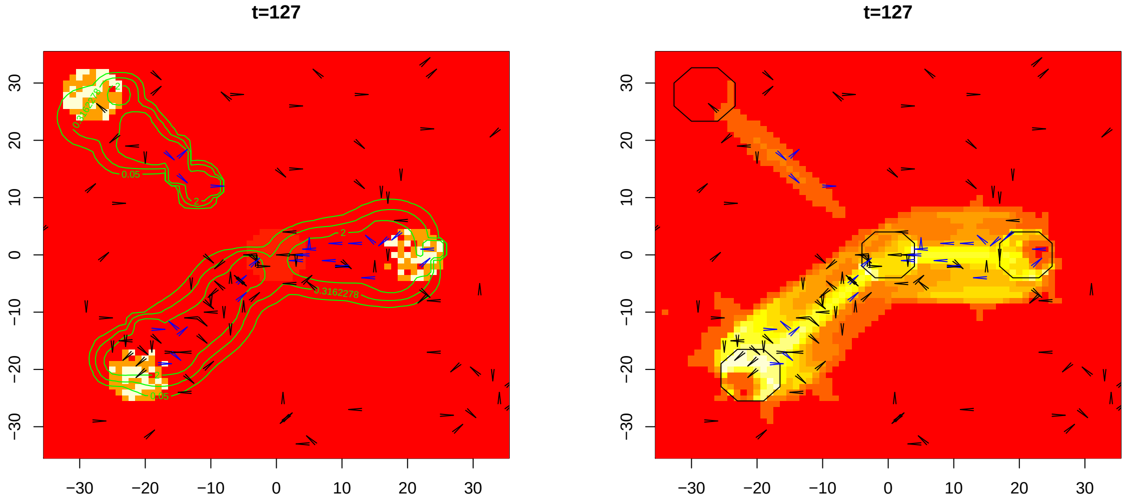
The **system state** is a Cartesian product of the states of each ant, which can occupy any of the 71×71 squares, moving in any of eight directions, and be with food or without; of the amount of food on each grid point of the food sources; and of the amount of chemical on each grid point in the world: • **ant state** $a\,(\text{ant}, x, y, \text{dir}, \text{food?}) \in \{0, 1\}$, where $x \in \{-35, \dots, 35\}$, $y \in \{-35, \dots, 35\}$, $\text{dir} \in \{\frac{0}{4}\pi, \dots, \frac{7}{4}\pi\}$, $\text{food} \in \{0, 1\}$ representing an ant without food and an ant with food respectively, and $\text{ant} \in \{1, \dots, 125\}$ indexes into an ant. • **food on grid** $f(x, y) \in \{0, \dots, \max_f\}$, where $x, y \in \{-35, \dots, 35\}$ indexes into a grid point under investigation. • **chemical on grid** $c(x, y) \in \{0, \dots, \max_c\}$, where $x, y \in \{-35, \dots, 35\}$ indexes into a grid point under investigation, and $\max_c$ is the cut-off amount of chemical that a grid point can hold (chosen according to computation time and inference precision). • **is this the nest?** $n(x, y) \in \{0, 1\}$, where 0 means that the grid point is outside of the nest, and 1 in the nest.

The **stochastic events** of the system include an ant loading food at a food source, an ant taking food from a food source, an ant unloading food at the nest, an ant leaving food at the nest, chemical diffusion, and chemical evaporation: • **ant unloading food** an ant with food drops it at the nest and leaves the nest, $a\,(\text{ant}, x, y, \text{dir}, 1) \wedge n(x, y) \rightarrow a\,(\text{ant}, x, y, \text{dir} + \pi, 0)$. • **ant loading food** an ant without food takes food at a food source and turns back to the nest, $a\,(\text{ant}, x, y, \text{dir}, 0) \wedge f(x, y) > 0 \rightarrow a\,(\text{ant}, x, y, \text{dir} + \pi, 1) \wedge f(x, y) - 1$. • **ant seeking food** $a\,(\text{ant}, x, y, \text{dir}, 0) \wedge f(x, y) \equiv 0 \rightarrow a\,(\text{ant}, x', y', \text{dir}', 1)$, an ant without food wiggles towards areas with high chemical concentration, where $(x', y') = \text{round}(x + \Delta t \cdot (\sin, \cos)(\text{dir}'))$, and we approximate this forward-wiggling behavior as defined in NetLogo language with $p(\text{dir} \rightarrow \text{dir}' = \text{dir} \pm (\sin, \cos)\Delta d) \propto c \cdot 1_{\Delta d \equiv 0} + f((x, y) + (\cos, \sin)\text{dir}')^{\gamma}$. • **ant heading to the nest** $a\,(\text{ant}, x, y, \text{dir}, 1) \wedge \text{not } n(x, y) \rightarrow a\,(\text{ant}, x', y', \text{dir}', 1)$, an ant with food wiggles towards the nest. • **chemical diffusion** $c(x, y) \wedge c(x + \Delta x, y + \Delta y) \rightarrow (1 - \delta) \cdot c(x, y) \wedge c(x + \Delta x, y + \Delta y) + \delta \cdot c(x, y)$, where $\Delta x, \Delta y \in \{0, \pm 1\}$.

Next, we decompose the system state into ant states, food states, and chemical states, and assume mean-field interactions among the ants, food, and chemical in order to circumvent tracking an astronomical number of states in the state space ($(71 \times 71 \times 8 \times 2)^{125} \times \max_f^{71 \times 71} \times \max_c^{71 \times 71}$). The amount of food on each grid point is a categorical distribution taking values from $1, \dots, \max_f$. The amount of chemical on each grid point is a categorical distribution taking values from $1, \dots, \max_c$. Each ant can be at any of the 71×71 grid points, moving in any of the eight headings, with or without food. The average amount of food to be taken by an ant at a grid point is the average amount of food at the grid point times the probability that an ant without food is on this grid point. The average amount of chemical to be left by an ant at a grid point is the probability that the ant with food is on this grid point. The chemical diffuses and evaporates according to a compound Poisson process.

With the mean-field inference algorithm, we can infer how ants in their initial positions at time 0 could move to their positions at time 200 based on their interactions with both chemical and food (Figure 2a), and on how other simulations could move the ants from the same origins to the same destinations with the same number of steps. We compare ant locations in a simulation with our inference from the first frame and final frames in Figure 2b, where different colors in the heat map represent the $\frac{x}{12}$ percentiles, where $x \in \{0, \dots, 12\}$. This animation demonstrates visually that our approximate inference has captured the dynamics of the model.

To assess the accuracy of the approximate inference on the Ants model, we independently sampled the time series of ants, chemical, and food from their respective hidden Markov processes under the mean-field approximation with the forward filtering backward sampling (FFBS) algorithm, and used the Metropolis-Hastings algorithm to accept or reject samples according to the exact probability of the Ant model. We set burn-in period and thinning interval heuristically, and evaluated convergence by comparing various statistics of independent Metropolis-Hastings runs. We found the probability of a sample path under the mean-field approximation to be a good estimation of the exact probability induced by the Ants world ($R^2 = 0.99, p < 10^{-6}$), and the marginal one-slice statistics to be a good estimation of those computed



(a) sample

(b) inference

Figure 2: Online version shows a simulation run of the Ants model (left) and the stochastic inference of ant trajectories from the end points (right). Printed version shows ant states (left) and inferred ant locations (right) at time 127.

from the Metropolis-Hastings sample ($R^2 = 0.95, p < 10^{-3}$). Hence, we believe the approximate inference developed in this paper has good accuracy.

The time/space complexity of the approximate algorithm scales linearly with the number of agents or system variables, and for each agent or system variable the complexity scales linearly with the number of latent state (because the state transition matrix induced by productions is sparse). The approximate forward-backward algorithm couldn't get noticeable improvement after 4 iterations, and the approximate forward-backward + parameter estimation algorithm couldn't get noticeable improvement after 20 iterations for a wide range of initial configurations. Hence the approximation based on greedy local projection is scalable to much larger data set.

While this example makes a stochastic inference about model-generated data assuming the model is correct, there is no restriction in making stochastic inferences about real-world data with competing agent-based models. Such a combination of model and data helps data scientists to tell stories about data and to identify a data business as an optimization problem on data, and also helps modelers to benchmark competing models against real-world data and to make predictions about real-world data with their models. For example, it could be interesting to combine traditional agent-based road-traffic models with petabyte vehicle tracking data and make predictions about events that have never previously occurred. Or, one could combine meme-diffusion models with real-world mobile data and web surfing data to predict the next fad, or combine city growth models with demographics to predict economic trends.

5 CONCLUSIONS AND DISCUSSIONS

In this paper, we gave a mean-field inference algorithm to approximate a continuous-time coupled Markov process defined by multi-agent models. Such inference algorithm is useful in bridging together big data and the world of generative models describable with rule-based production systems. Statistics such as stopping times about an individual chain can be estimated from fitting Markov models if we employ structured variational approximation. Such variational algorithms decouple the interacting chains, and so cannot be used to estimate statistics about the interactions among chains; however, in such situations estimations can be based on Monte Carlo algorithms.

REFERENCES

Anderson, J. R. 1996. *The architecture of cognition*. Mahwah, N.J.: Lawrence Erlbaum Associates.

Batty, M. 2007. *Cities and Complexity: Understanding Cities With Cellular Automata, Agent-Based Models, and Fractals*. Mit Press.

Castellano, C., S. Fortunato, and V. Loreto. 2009. “Statistical physics of social dynamics”. *Reviews of Modern Physics* 81 (2): 591–646.

Dong, W., K. A. Heller, and A. Pentland. 2012. “Modeling Infection with Multi-agent Dynamics”. In *SBP*, edited by S. J. Yang, A. M. Greenberg, and M. R. Endsley, Volume 7227 of *Lecture Notes in Computer Science*, 172–179: Springer.

Dong, W., B. Lepri, and A. Pentland. 2011. “Modeling the co-evolution of behaviors and social relationships using mobile phone data”. In *MUM*, edited by Q. Dai, R. Jain, X. Ji, and M. Kranz, 134–143: ACM.

Dong, W., and A. Pentland. 2009. “A Network Analysis of Road Traffic with Vehicle Tracking Data”.

Dudley, R. M. 2002. *Real analysis and probability, 2nd ed.* Cambridge University Press.

Epstein, J. M. M. 2007. *Generative Social Science: Studies in Agent-Based Computational Modeling (Princeton Studies in Complexity)*. Princeton University Press.

Goodwin, R. M. 1975. *Socialism, capitalism and economic growth*, Chapter A growth cycle. Cambridge University Press.

Heskes, T., and O. Zoeter. 2002. “Expectation Propogation for Approximate Inference in Dynamic Bayesian Networks”. In *UAI*, edited by A. Darwiche and N. Friedman, 216–223: Morgan Kaufmann.

Ibe, O. C. 2008. *Markov processes for stochastic modeling*. Academic Press.

MATSim development team (ed.) 2007. “MATSIM-T: Aims, approach and implementation”. Technical report, IVT, ETH Zürich, Zürich.

MIT Human Dynamics Lab 2012. “http://realitycommons.mit.edu Reality Commons”.

Newell, A., and H. A. Simon. 1972. *Human problem solving*. Englewood Cliffs, N.J.: Prentice-Hall.

Orange S. A. “http://www.d4d.orange.com/ Data for Development”.

Popper, K. 1965. *Conjectures and refutations: the growth of scientific knowledge*. London: Routledge and Kegan Paul.

Smith, L., R. Beckman, and K. Baggerly. 1995. “TRANSIMS: Transportation analysis and simulation system”. Technical report, Los Alamos National Lab., NM (United States).

Wainwright, M. J., and M. I. Jordan. 2008. “Graphical Models, Exponential Families, and Variational Inference”. *Foundations and Trends in Machine Learning* 1 (1-2): 1–305.

Uri Wilensky 1997. “NetLogo Ants model”. http://ccl.northwestern.edu/netlogo/models/Ants. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilkinson, D. J. 2006. *Stochastic modelling for systems biology*. Boca Raton, FL: Taylor & Francis.

AUTHOR BIOGRAPHIES

Wen Dong is an Assistant Professor of Computer Science and Engineering at the State University of New York at Buffalo with a joint appointment in the Institute of Sustainable Transportation and Logistics. He focuses on modeling human interaction dynamics with stochastic process theory through combining the power of “big data” and the logic/reasoning power of agent-based models, to solve our societies most challenging problems such as transportation sustainability and efficiency. Wen Dong holds a Ph.D. in Media Arts and Sciences from Massachusetts Institute of Technology. His email address is wendong@buffalo.edu.