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.

PARTICLE FILTERING USING AGENT-BASED TRANSMISSION MODELS

Kurt Kreuger
Nathaniel Osgood

Department of Computer Science
Computational Epi & Public Health Informatics Lab
110 Science Place
Saskatoon, SK S7N 5C9, CANADA

ABSTRACT

Dynamic models are used to describe the spatio-temporal evolution of complex systems. It is frequently difficult to construct a useful model, especially for emerging situations such as the 2003 SARS outbreak. Here we describe the application of a modern predictor-corrector method – particle filtering – that could enable relatively quick model construction and support on-the-fly correction as empirical data arrives. This technique has seen recent use with compartmental models. We contribute here what is, to the best of our knowledge, the first application of particle filtering to agent-based models. While our particle models adapt to different ground-truth conditions, agent-based models exhibit limited adaptability under some model initializations. Several explanations are advanced for such behavior. Since this research serves as an initial foray into this line of investigation, we draw out a clear path of the next steps to determine the possible benefits of using particle filters on agent-based models.

1 INTRODUCTION

Epidemiology seeks to understand the dynamics of disease spread through human and animal populations. In spite of advances in public health knowledge over the last century, infectious disease still places a large burden on length and quality of life. Traditional epidemiological study makes heavy use of time-series data (Osgood and Liu 2014) and regression analysis techniques (Auchincloss and Diez Roux 2008). Public health agencies will use this data to identify possible causal relationships between health-related parameters. Such approaches provide one way for making predictions about evolution of the burden and distribution of disease in a population. Unfortunately, such approaches tend to scale poorly to systems exhibiting hallmarks of complex systems, such as non-linearity, reciprocal causality, feedback loops, and resulting phenomena such as tipping points and other threshold effects, lock-in, etc. Such features are particularly notable in the context of communicable disease spread.

Dynamic modeling is an approach that leverages multiple types of domain-specific data to help clarify and account for multiple causal pathways. Lipsitch et al. (Lipsitch et al. 2003) constructed a conceptually simple model early on in the 2003 SARS outbreak to approximate the basic reproductive number.

Dynamic modeling does have its own limitations, however. Models need to be sufficiently calibrated to real-world data to be able to make actionable predictions. While roughly calibrated models can often be employed to make short-term predictions, over time the discrepancy between model predictions and empirical observations will increase, requiring further calibration (Osgood and Liu 2014). This growing discrepancy reflects inaccurate parameter values, omitted or oversimplified characterization of dynamic processes, and the vagaries of stochastic fluctuations in empirical processes. The resulting model can be unreliable and costly. In the control of communicable illness outbreaks, there has traditionally not only been a heavy emphasis on use of continuous data from the field, but a strong recognition of the value of quickly built models that give actionable insight.

Predictor corrector models have been in common use for the better part of a century. One of the oldest and most widely applied is the Kalman filter (KF). In addition to extremely broad applications in real-time control context, this approach has been applied successfully in the context of communicable diseases models (Qian, Osgood, and Stanley 2014). However, Kalman filtering is associated with several notable shortcomings. Firstly, it relies on linear models or linearized versions of non-linear models in the case of the Extended Kalman Filter (Osgood and Liu 2014). Coupled with the KF's reliance on a maximum likelihood estimate (MLE), such linearization can pose tremendous challenges in the context of the non-linear models typical in communicable disease epidemiology. Secondly, KF presupposes Gaussian distributed errors for both measurement error and process noise – an assumption that can be highly problematic in the case of epidemiological count reporting processes, such as those with low numbers. Finally – and most importantly for this research – the KF linearization assumption cannot be readily applied to agent-based models, whose dynamics do not readily admit to representation using continuous-time differential equations.

Recent papers have leveraged particle filtering in the context of compartmental models (Ong et al. 2010; Osgood and Liu 2014). The idea is to use a compartmental model that incorporates broadly accurate dynamics for a given complex system. Then a particle filter algorithm is implemented over many iterations of the model. Using the weight-update and resample procedures of the particle filter, the collection of particles is hoped to approximate the posterior distribution over states and parameters over time. In practice, this can be used to enable to construction of quick and relatively simplified models which, when combined with real-world data, develop the ability to provide predictive value.

Indeed, in (Ong et al. 2010), the authors built a compartmental SIR model for a real-time H1N1 outbreak in Singapore, using data that came in from general practice and family doctor clinics throughout the country; they found that, in spite of some significant model over-predictions, and some significant model simplifications, the disease peak was well predicted. Also, the farther along the epidemic cycle, the more accurate the predictions of the particle filter became. The authors describe how particle filtering can be a powerful tool in building relatively simple models, which can be done in a more timely fashion, but still enabling those models to provide predictive ability for an ongoing and new epidemic outbreak.

In (Osgood and Liu 2014), the authors took a more theoretical approach, and built an agent-based model of a simple disease progression to serve as synthetic ground truth. A compartmental model was constructed using similar disease assumptions. The different natures of agent-based and compartmental models and systematic inaccuracies in a parameter value resulted in significantly different predictions of new disease infections over time. However, when using the particle-filtering technique, the compartmental model collection was able to strongly reduce the errors inherent in compartmental models.

In contrast to such previous contributions, this research applies this technique on an agent-based particle model rather than the compartmental models. This is important for two reasons. Firstly, the particle filter – in contrast to the long-standing Kalman Filter (Qian, Osgood, and Stanley 2014) – does not assume a particular mathematical formulation of model dynamics. Secondly, there are areas where agent-based models have certain key advantages over compartmental models. Agent-based models are more able to represent the heterogeneous nature of agents – notably including heterogeneous histories – and their social networks (Tian and Osgood 2011).

This research marks what we believe to be the first application of particle filtering to ABMs.

2 ALGORITHM DESCRIPTION

2.1 Particle Filter Intuition and Derivation

To understand how a particle filter is implemented here, it is helpful to understand the stochastic nature of the dynamic models being employed to understand an unknown ground-truth. In the communicable disease area, an agent-based model seeks to simulate processes such as contact, transmission over such a contact, incubation, recovery, and waning of immunity. Some sources of stochastics will be due to the need to capture temporal variability of a highly unknown parameter. How often or for how long does a person contact another person in their social network? That might best be modelled by drawing from a probability distribution.

A second type of stochastics comes from the need to generalize model predictions. If we could measure the social network of a population exactly, then we could make predictions about that particular population, but it would not be easily generalizable to others. So we would use a network type, which has basic structures in common but whose exact connections vary from simulation to simulation.

Thirdly, some characterizations of model predictions as stochastic reflect the fact that for certain processes, statistical regularities may be better understood than the mechanics of that process.

The implications of the above is that there are uncertainties about the model state from time-step to time-step. At the same time, there will be incoming empirical observations y_t. The goal, therefore, is to find a way of sampling from the distribution $p(X_0, X_1, X_3, \dots X_t | y_0, y_1, y_2 \dots y_t)$ over model state at a given time, taking into account both uncertainty in model predictions and that associated with observation error. By sampling over this distribution, we can – by extension – then sample over future trajectories of the model, sampling from the difference in gains between baseline and an intervention, or between two such interventions, and compute the probability of different events (e.g., the probability that a given intervention will save more than a certain count of lives, or will be cost saving).

The particle filter provides a means of accomplishing such sampling. The key challenge associated with this approach that of sampling from this system state distribution $p\left(X_0^{(i)}, X_1^{(i)}, X_2^{(i)}, \dots X_T^{(i)} \middle| y_0, y_1, y_2 \dots y_t\right)$. This challenge is particularly acute given that for non-linear models, we lack any closed-form characterization of the distribution. Particle filtering provides an approach of sampling from the distribution by sampling from many weighted particles over model state, each associated with a specific state vector at a given time. More specifically, in accordance with the approach of importance sampling, we approximate sampling from such an unknown distribution over model states by sampling from a readily available distribution (a proposal distribution), but weighting the samples according to their relative probability (density). More formally, importance sampling states that sampling from a target density $b(x)$ can be approximated by sampling from another distribution – termed the proposal distribution, or importance proposal – q(x), attaching a weight to that sample equal to $\frac{b(x)}{q(x)}$, and then sampling in turn from those samples, with a probability equal to their weight. In this case, each of those samples from the distribution is termed a particle.

In the case of the particle filter, the weight updates are performed only at points of observation and are computed recursively, with the weights just prior to a given observation being dependent on the weight after the previous observation, and those weights only being updated at the point of the observation. For notational simplicity, we first consider the case of a single particle associated with state X_t at time t. That is, we decompose

$$p(X_0, X_1 X_2, \dots X_t | y_0, y_1, y_2 \dots y_t) = p(X_t | y_t, X_{t-1}) p(X_{t-1} | y_{t-1}, X_{t-2}) \dots p(X_1 | y_1, X_0).$$

To understand the weight update rules to allow us to sample from the above, it is valuable to understand the components that will be used. We denote the state of the model at time t for particle i as $X_t^{(i)}$. The likelihood function $g\left(y_t \middle| X_t^{(i)}\right)$ specifies the likelihood that we will observe the empirical data y_t at time t given the model state $X_t^{(i)}$ posited by particle i at time t. This likelihood will be specified explicitly by the modeler. The probability density $p\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)$ specifies the probability density of a particle arriving at state $X_t^{(i)}$ at time time t, given that that particle was in state $X_{t-1}^{(i)}$ at time $t-1$. This probability density is implicit in light of the emergent and stochastic behavior of the system under study.

To sample from $p(X_1|y_1,X_0)$ via particle filtering¸ we note that we require a weight w_1 equal $\frac{p(X_1|y1,X_0)}{q(X_1|y_1,X_0)}$, and that for $t \geq 1$,

$$w_t = \frac{p(X_t|y_t,X_{t-1})}{q(X_t|y_t,X_{t-1})} w_{t-1}$$

We further note that

$$\frac{p(X_t|y_t,X_{t-1})}{q(X_t|y_t,X_{t-1})} = \frac{p(X_t|X_{t-1})g(y_t|X_t)}{q(X_t|y_t,X_{t-1})}$$

The final step is to realize that we can set our importance proposal to simplify this equation. If we choose $(X_t|y_t,X_{t-1}) = p(X_t|X_{t-1})$, (i.e., if we choose not to consider the empirical data y_t in evolving the model between observations), we have

$$\frac{p(X_t|X_{t-1})g(y_t|X_t)}{q(X_t|y_t,X_{t-1})} = g(y_t|X_t)$$

and we only need to specify the likelihood g. We follow (Osgood and Liu 2014) an employ a negative binomial distribution.

$$g(y_t|x_t) = \binom{y_t+r-1}{y_t} p^{y_t}(1-p)^r \tag{4}$$

where $p = \frac{x_t}{x_t+r}$ and can be interpreted probability that a given reported case is a true case. r is the dispersion parameter. Because the distribution $g(y_t|x_t)$ will in general assign a non-zero likelihood to both the cases where the empirical data is greater than or equal to the actual model state, this distributional assumption helps avoid a pathology in which all particles are assigned a likelihood of 0.

Every so often, the particle population will need to be resampled. This is due to the fact that, as the particle filter progresses, a few particles might have high weights while most others will have weights near 0. The advantage of simulating many particles will be diminished given that most predictive power will come from those few highly-weighted particles. In this case, therefore, a fresh batch of particles will be created by sampling from the particles according to their weights and creating copies. Most new particles will be copies of the fewer highly-weighted originals. Weights are then renormalized. While not explored here, sampling from the entire trajectory of states in the context of resampling requires tracing the lineage of a given particle from the first to the last time point, and then sampling from the particles at the final time according to their weight.

2.2 Final Algorithm

As can be seen from the section above, there are 2 basic steps: the first to set up the particle filter, and the next to iterate.

1. Initialization step: where time $t = 1$, for $i \epsilon \{1, N\}$
   a. Sample $X_1^{(i)}$ from $q_1(x_1|y_1)$
   b. Calculate an initial weight for each particle, assuming uniform weights $w_1^{(i)} = \frac{1}{N}$.
2. Update step: where time $t \geq 2$
   a. Sample $X_t^{(i)} \sim q_t\left(x_t \middle| y_t, X_{1:t-1}^{(i)}\right)$
   b. set $X_{1:t}^{(i)} = \left(X_{1:t-1}^{(i)}, X_t^{(i)}\right)$
   c. Update weights using:
      $w_t^{(i)} = w_{t-1}^{(i)} g\left(y_t \middle| X_t^{(i)}\right)$
   d. Normalize weights to sum to 1
   e. Resample

3 GROUND TRUTH MODEL

3.1 Introduction

We describe here the ground-truth model that generated the data for our particle filtering algorithm. The ground-truth model was constructed using the AnyLogic framework (based on Java), while, for several design reasons, the particle model was constructed in pure Java and implemented within the AnyLogic project. The entire modeling framework was thus a hybrid one from two different perspectives. Firstly, the framework combined agent-based modeling methodology with a particle filtering methodology in a novel manner. Secondly, the model exhibited a hybrid handling of time, with the ground truth model running in continuous time using AnyLogic's agent-based modeling mechanisms, and the particle filter versions of the models running in discrete time in custom Java classes.

3.2 Ground Truth Model

The structure for the ground truth model adheres to the extensively applied SEIR modeling paradigm (Osgood and Liu 2014). Agents in themselves are categories as being in one of 4 states: (S)usceptible, (E)xposed, (I)nfective, (R)ecovered. (Colditz et al. 1994) (This approximates many diseases of importance, including SARS, since infected individuals go through latent state before becoming infective themselves. See (Lipsitch et al. 2003).) One clear benefit of this approximation is that it can be applied with agent-based models as undertaken here and in (Osgood and Liu 2014), or in a compartmental ground-truth model as employed in (Ong et al. 2010). While not done here, this can facilitate the comparison of predictions between agent-based and compartmental models.

Within the ground-truth model, agents are placed inside of a social network, and agents have the ability to exchange messages only between other agents with whom they are connected. As discussed in the next section, the structure of this network varies from scenario to scenario.

3.2.1 Network Type

This work used four prominent network types: distance based, Poisson Random ("random"), Ring Lattice, and Watts-Strogatz small world ("small world"). The random network requires the specification of an average number of connections per agent; each pair of individuals in the population is connected with equal probability in light of the implied network density. In AnyLogic, the network choice was declaratively specified (and used AnyLogic's built-in support for those explored network types), whereas it was constructed manually in the particle model.

Secondly, we employed a distance-based network. Here, we specified a connection distance threshold. A given pair of agents is connected if and only if they lie within a specified distance of each other. This network depends importantly on the spatial layout assigned to the agents. In both particle and ground-truth models, both the X and Y coordinates of agents were drawn independently from continuous uniform distributions between 0 and 500. A third network type used was a ring lattice network, in which all nodes are logically placed in a ring, connecting nodes with are within n nodes in either direction. Finally, the small world network represents a mixture between a ring lattice and a random network. The small world network can be generated by starting with a ring lattice network, and then a specified fraction of the connections are rewired to be connected with another node picked with uniform probability from across the population.

Several features distinguish the networks. While all serve as conduits for stochastic percolation processes for pathogen, the structure of a random network is less affected by these stochastics. In other words, many different random networks, with the same number of connections and the same number of nodes lead to very similar connection patterns (because the variation of the number of connections for a given node is approximated by a Poisson distribution (Erdos and Renyi 1960)), and hence, similar behavior of disease spread.

On the other hand, distance based networks can be quite different between versions using the same construction parameters. If the connection distance is not so large as to make nearly everyone connected, distance networks can have disconnected subnetworks. The number and size of these disconnected sections can vary between parametrically similar networks. Even where the entire network consists of a single component, the highly overlapped nature of the paths from a given node (e.g., an infective) to other nodes means that the spread of contagion across such paths can be far more readily "blocked" than is the case in random networks.

By contrast, small world networks exhibit a mixture of the locality exhibited by ring lattice networks and the globally interconnected structure of random networks. As discussed below, this difference between networks under stochastics will affect how the particle models are able to adapt to different conditions.

3.2.2 Agent Behavior

In the ground-truth model, agents decide their behavior according to the state-chart indicated in Figure 1. Here, agents all start off in the susceptible state. They can become exposed only when receiving an infecting message from an agent currently spreading the disease to whom they are connected. Once exposed, they will wait for a precisely defined incubation period, a parameter of the model, before becoming infective. For every time unit an agent spends in the infective state, it performs an action to expose a single neighbor to infection; if that neighbor is susceptible, the neighbor will enter the exposed state. Finally, after a precisely defined recovery period (another model parameter), that agent becomes recovered, which means that they do not infect others, and cannot become infected by others.

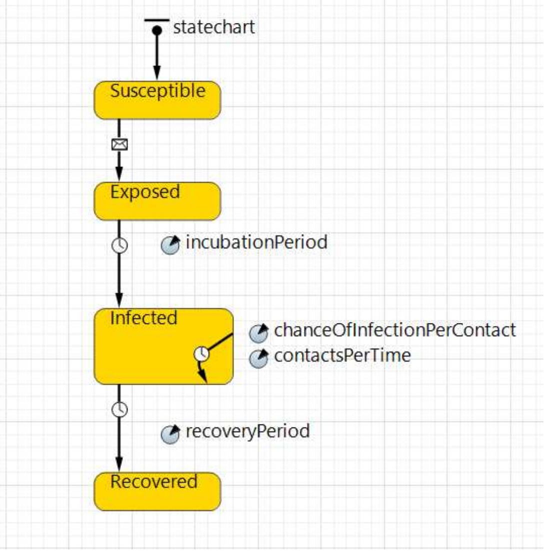

Figure 1: Agent state chart in the ground-truth model.

3.2.3 Actual and Reported Data

Following (Osgood and Liu 2014), within the ground truth model, we considered a difference between the actual daily incidence rate and the reported incidence rate. When someone gets sick, we assume a 50% chance of reporting that illness, with the reported count being drawn from a binomial distribution. The particle weights are updated according to the likelihood of observing that reported incidence (y_t).

3.2.4 Evaluation Metric

In order to assess the effectiveness of the particle filter in improving model accuracy, we compared the performance of the model absent particle filtering with that which included particle filtering. The comparisons were performed over 24 realizations and used a sum squared error metric to judge the discrepancy between the ground truth and estimates in sampled particles (Osgood and Liu 2014).

Specifically, at each timestep t, prior to any particle weight update, we computed a discrepancy for a given particle position (sample) by sampling from 1000 particles according to their weights; for each such particle, we computed the square of the difference between that particle and the ground truth value. We then summed the average of the square of those values evaluated across all of the sampled particles.

4 METHODS AND RESULTS

Figure 2 displays the outputs from each of 9 Monte Carlo simulations we ran; for each of the Random, Small World and Distance based network types, we ran a simulation with the particle off, every time step, and every 5 time steps. All of these networks were structured so as to have, on average, approximately 10 connections per agent.

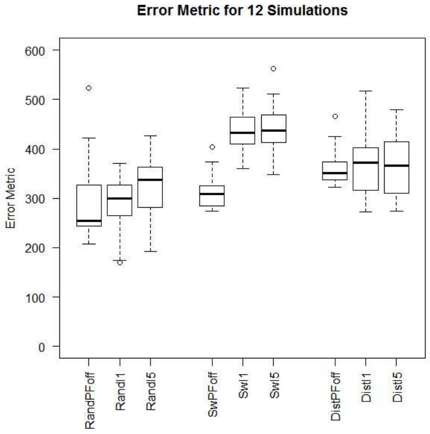

Figure 2: Box plot of error metric over 9 simulations using 3 network types (random, small world, distance-based) with 3 particle filtering setups (no weight updating, updating every 1 time step, and updating every 5 time steps).

What is shown in Figure 2 is that the particle filter actually results in poorer performance of the particles, most especially for the small world network. For this reason, we ran 6 more simulations, presented in Figure 3.

In Figure 3a we used a distance network, but characterized a situation in which the modeler was uncertain of the exact connection distance used in the ground truth model. As a result, while the ground truth model makes use of a specific connection threshold (41.5), the particles are created using a connection threshold drawn from a uniform distribution from 20 to 60 (runs are shown with “dist” appended to the name). As shown in Figure 3a, the particle filter without particle weight updating (D_PFoff_dist) returns an average error of 429 with standard deviation of 20.3. Updating each time step (D_I1_dist) reduces the mean to 344, but increases the deviation to 96.8. The Kolmogorov-Smirnov (KS) test between D_PFoff_dist and D_I1_dist is $r_{KS} < 0.0001$. Using an update interval of 5 (D_I5_dist) reduces the spread, but returns $r_{KS} = 0.14$ in relation to without weight update. Clearly, in the case of uncertain particle parameters, the particle filter is results in a beneficial effect on the predictive ability of the particles.

In Figure 3b, we use a ring lattice network (implemented in the particles as a small world network with $\beta = 1.0$), and employ 640 realizations. Similar to the varied-parameter distance network, using an update period of 1 time step significantly drops the mean (73.3 to 66.0) and increases the standard deviation (1.71 to 7.43). However, the update period of 5 time steps returns almost the same output as no update period at all (mean of 73.8 and deviation of 1.36).

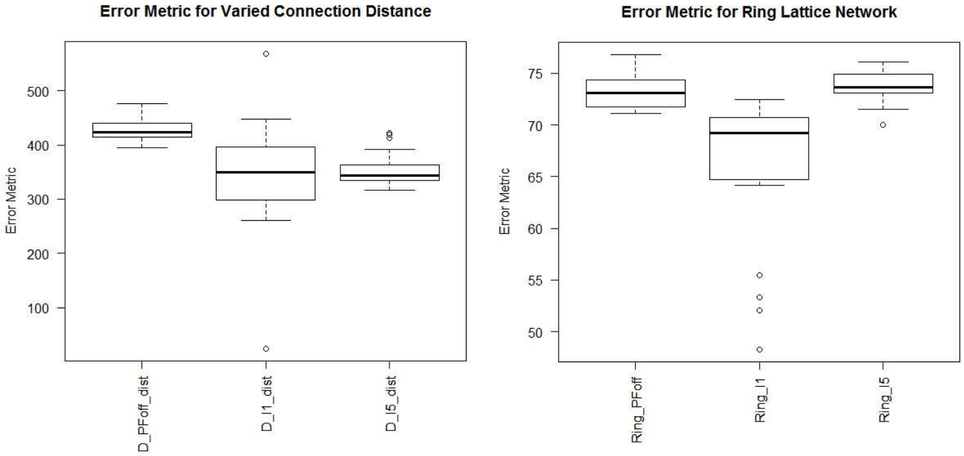

Figure 3: a) Box plot of error metric over 3 simulations using a distance-based network with ground truth connection distance of 41.5 and particle connection distance drawn from a uniform distribution from 20 to 60. Results shown for no weight update period, 1 time step updates, and 5 time step updates. b) Box plot of error metric over 3 simulations using a ring-lattice network with connection count of 2 per agent. Results shown for no weight update period, 1 time step updates, and 5 time step updates.

Finally, Figure 4 displays the results from two simulations using a ring lattice network with 640 realizations (as opposed to the 24 realizations for all other results). With an $r_{KS} < 0.0001$, it is clear that the presence of the particle filter in Ring_I1_640 provided a better fit to ground-truth output.

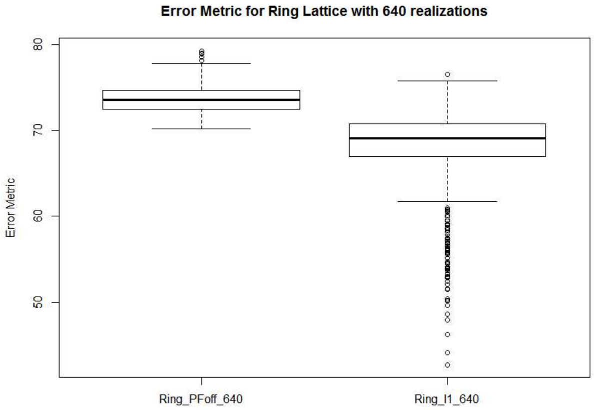

Figure 4: Box plot of error metric over 2 simulations using a ring-lattice network.

5 DISCUSSION

The basic format of this experiment is mathematically similar to (Osgood and Liu 2014). We initially expected to see the same type of particle filter output. That experiment produced Figure 5. There are two especially noteworthy features. The first is that there was exhibited a wide range of particle trajectories often near the ground-truth data, expanding wider throughout the time between weight updates (indicated by the red and purple horizontal lines).

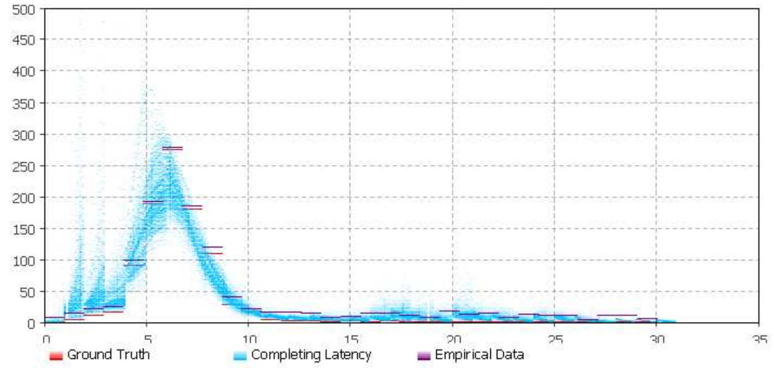

Figure 5: Similar particle filter using a compartmental particle model (Osgood and Liu 2014).

A second finding from (Osgood and Liu 2014) is that the particle filter improves accuracy throughout the model. The range of the particle predictions (in blue) is very broad early in the model run. The discontinuities visible at $t = 2$, and 3 result from particles being reweighted and resampled. An example output from this particle filter is shown in Figure 6. These two features -- very prominent in (Osgood and Liu 2014) -- stand in contrast to the results here.

Our initial results in Figure 2 bear this out. The agent-based particle filter either increases the mean error, increases the deviation, or both. Figures 3 and 4, however, provide strong evidence that the particle filter logic is properly formulated in the model, and that the observed behavior differs strikingly between different scenarios. This was further borne out by an extensive examination of model logic that drew on much previous familiarity with successfully operating particle filters.

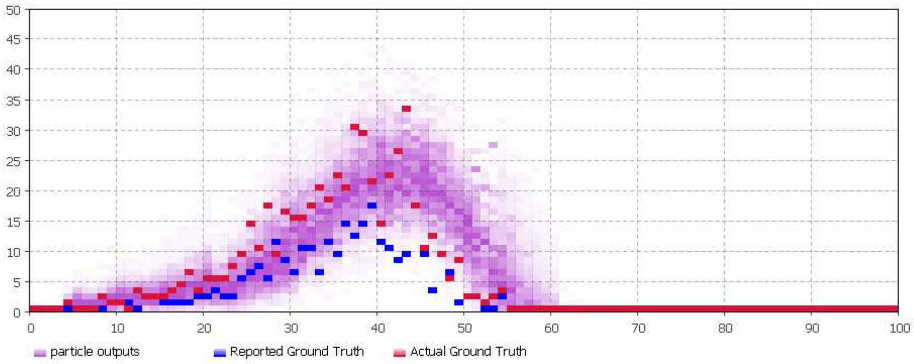

Figure 6: Example output for small world network ($\boldsymbol{n} = \mathbf{10}, \boldsymbol{\beta} = \mathbf{0.9}$). Weight update occurs every 10 time steps starting at $\boldsymbol{t} = \mathbf{1}$.

We can think of two possible explanations for the poor performance of the particle filter in Figure 2 and its improved performance in Figures 3 and 4. The first is that the ground-truth model and the particle model are logically identical. By contrast, in (Osgood and Liu 2014), the ground-truth model is an ABM whereas the particles employ an aggregate model. They are necessarily different from each other, with the particle models inevitably diverging from the ground truth.

By contrast, in the current investigation, the simulations where the weight-update cycles are turned off constitute the best possible modeling situation —a situation where we are employing a model that precisely captures the dynamics of the underlying system being studied. The failure of the particle filter to improve upon the results of the “open loop” model with the particle filter off may be simply be reflection of the difficulty of improving upon a highly accurate characterization of the underlying system.

To probe this situation, we constructed Figure 3, which assumes a modeling context in which there is imperfect knowledge of the connection length in the distance-based network. Introducing this source of uncertainty – and this mismatch between the dynamics of the ground truth model and that resulting from the particle models – demonstrates strong benefits of particle filtering. As to why the update cycle of 5 time steps offers a smaller dispersion, it is currently assumed that this is because it would be filtering particles based on a 5-time-step sum of incidence cases. Small variations in the ground-truth output (whether due to actual incidence cases or the reporting of cases) are smoothed out, leading to fewer good candidate particles being filtered out because of momentary disagreement with ground-truth output.

The second possible contributor to the failure of the particle filter to secure gains in accuracy for the scenarios depicted in Figure 2 is that the agent-based particles have a much higher dimensionality than the aggregate particles from (Osgood and Liu 2014). This leads to two complexities. Firstly, increasing the state-space dimensionality increases the required particle count to obtain an adequate sampling density from the state-space. Secondly, the particle fitness metric (incidence cases), which is used to compare particles, is a highly aggregate model output that confers comparatively little information by which to select one particle over another. In essence, there could be many local minima in the state-space where models of vastly different internal structure might nevertheless compare favorably using the simple fitness metric of incident cases.

These hypotheses are lent credibility by Figure 4. By virtue of its use of a deterministic network, these scenarios result in the ground-truth and particle models all having the same network connections. There will still be stochastics involved (e.g., regarding the timing of agent infection), but such a model exhibits drastically smaller state-space dimensionality than do those examined in the earlier figures. Given the lower dimensionality of the state space, the 1000 particles are thereby posited to occupy a much higher state-space density, with a correspondingly improved set of possible particles for selection. Figure 4 demonstrates that running the update every time step noticeably improves particle filter performance over the no-update "open loop" version. As to why the 5-time-step update does not improve performance, it could be simply that the update frequency is too low. More simulations need to be run to examine this particular issue.

Connected with this increase in state-space dimensionality is the increased computational complexity of the agent-based particle filter. While the high dimensionality of the agent-based model would make highly desirable very large ensembles (e.g., 1,000,000) of particles, the per-particle computational cost makes that infeasible. By contrast, aggregate particle filters require less large ensembles (due to lower dimensionality), and can more readily support them computational. An open question is what is the gain from using agent-based particle filters with fewer particles vs using aggregate particle filters with many more particles?

There are several other notable complexities about this particle filtering algorithm in general. The ground-truth model uses a binomial draw to determine how many actual cases are reported. This reported number is what is used in the likelihood function of the particle in eq. (4). The dispersion parameter essentially determines how broad or narrow the filter is. Larger r values make the negative binomial more peaked, and therefore admits a narrower band of particles "theories" as to the current state. Hence, this parameter has an effect on the whole particle filter behavior. Finding an appropriate value for r might improve performance.

There is a question as to the very idea of using a negative binomial in the likelihood function. Indeed, a binomial distribution makes more intuitive sense (as a chance of "success" or "failure" – reporting or not – where each opportunity is independent of all others). However, the binomial distribution has the unfortunate quality where particles positing a count of infectives less than the reported incidence are associated with a likelihood – and thus a weight – of 0. In situations where all particles posit more than the reported incidence, no particle weight can be calculated. This is the primary reason for selecting a negative binomial likelihood formulation.

6 SUMMARY

This research provides the first known application of particle filtering to particles corresponding to agent-based models. We found that the particle models were able to adapt to compensate for mismatches between the model characterization of system dynamics and the actual dynamics, but observed that this adaptation encountered surprising limits in compensating for stochastic fluctuations within the observed data in those cases where the models associated with the particles highly accurately depict the ground truth. We advanced here several possible explanations, for the observed limits, and propose additional lines of investigation responsive to these challenges.

REFERENCES

Auchincloss, A. H., and A. V. Diez Roux. 2008. “A New Tool for Epidemiology: The Usefulness of Dynamic-Agent Models in Understanding Place Effects on Health.” American Journal of Epidemiology. doi:10.1093/aje/kwn118.

Colditz, G. A., T. F. Brewer, C. S. Berkey, M. E. Wilson, E. Burdick, H. V Fineberg, and F. Mosteller. 1994. “Efficacy of BCG Vaccine in the Prevention of Tuberculosis. Meta-Analysis of the Published Literature.” *JAMA : The Journal of the American Medical* Association 271 (9): 698–702. http://www.ncbi.nlm.nih.gov/pubmed/8309034.

Erdos, P., and A. Renyi. 1960. “On the Evolution of Random Graphs.” Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5: 17–61.

Lipsitch, M., T. Cohen, B. Cooper, J. M. Robins, S. Ma, L. James, G. Gopalakrishna, S. K. Chew, C. C. Tan, M. H. Samore, D. Fisman, and M. Murray. 2003. “Transmission Dynamics and Control of Severe Acute Respiratory Syndrome.” Science 300 (5627): 1966–70. doi:10.1126/science.1086616.

Ong, J. B. S., M. I.-C. Chen, A. R. Cook, H. C. Lee, V. J. Lee, R. T. P. Lin, P. A. Tambyah, and L. G. Goh. 2010. “Real-Time Epidemic Monitoring and Forecasting of H1N1-2009 Using Influenza-like Illness from General Practice and Family Doctor Clinics in Singapore.” PLoS ONE 5 (4). doi:10.1371/journal.pone.0010036.

Osgood, N., and J. Liu. 2014. “Towards Closed Loop Modeling: Evaluating the Prospects for Creating Recurrently Regrounded Aggregate Simulation Models Using Particle Filtering.” In Proceedings of the 2014 Winter Simulation Conference, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 829–941. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Qian, W., N. D. Osgood, and K. G. Stanley. 2014. “Integrating Epidemiological Modeling and Surveillance Data Feeds : A Kalman Filter Based Approach.” Social Computing, Behavioral-Cultural Modeling and Prediction 8393: 145–52.

Tian, Y., and N. Osgood. 2011. “Comparison between Individual-Based and Aggregate Models in the Context of Tuberculosis Transmission.” In Proceedings, the 29th International Conference of the System Dynamics Society, 1–29. http://www.systemdynamics.org/conferences/2011/proceed/papers/P1356.pdf.

AUTHOR BIOGRAPHIES

KURT KREUGER is a doctoral student at the University of Saskatchewan. His email address is kurt.kreuger@usask.ca.

NATHANIEL OSGOOD is Associate Professor in the Department of Computer Science, Associate Faculty in the Department of Community Health and Epidemiology, and Director of the Computational Epidemiology and Public Health Informatics Laboratory. His email address is nathaniel.osgood@usask.ca.