Agent-Based Models: Statistical Issues and Challenges

David Banks

Department of Statistical Science

Duke University

1. History of Agent-Based Models (ABMs)

ABMs arise in computer experiments in which it is possible to define interactions "locally" for each agent, and then simulate emergent behavior that arises from the ensemble of local decisions.

Examples include:

- Weather forecasting, in which each agent is a cubic kilometer of atmosphere, and the local interactions are exchange of pressure, temperature, and moisture.
- Auctions, as in Yahoo! or Google, to determine which ads are shown to users.
- Traffic flow models, as in TRANSIMS, where agents (drivers) space themselves according to the actions of other drivers, and make route choices based on congestion avoidance.

Often the localization is geographic, but this is not essential. The auction example has no spatial component.

ABMs began in the 1940s, with ideas by von Neumann and Ulam relating to cellular automata (objects which, under a fixed set of rules, produce other objects, usually on paper grids—the most famous example is J. H. W. H. Conway's Game of Life).

This led to mathematical theories of interactive particle systems (Frank Spitzer, David Griffeath), which used methods from statistical mechanics to study problems related to phase changes and system dynamics.

As the agents developed more complex rules for interaction, and as the applications became more tuned to simulation studies of observable phenomena, the field morphed away from mathematics into economics, social science, and physics.

These models are popular because they enjoy a certain face-validity, and because they are often easy to program.

ABMs are computer-intensive, and so did not become widely used until the 1990s. A major impetus was Growing Artificial Societies, by Joshua Epstein and Robert Axtell (1996, MIT Press). This book showed how simple and interpretable rules for agents could simulate behavior that was interesting in demography, anthropology, sociology, and economics.

Their approach was to posit a **sugarscape**, a plane on which, at each grid point, "sugar" grew at a constant rate. Agents had locations on the sugarscape, and consumed the sugar at their location until it was exhausted, and then moved to a new location (in the direction of maximum sugar, without diagonal moves, with minimum travel and a preference to be near previous locations).

This simple system of rules led to circular migration patterns. These patterns are supposed to be similar to those observed in hunter-gatherer populations.

More rules created additional complexity. Epstein and Axtell added sex, and when there was sufficient sugar, the agents would reproduce. This led to age pyramids, carrying capacity, tribal growth with co-movement and fission, and other demographic features.

They added "spice", a second resource similar to sugar, and simple rules led to barter economies.

They added "tags", which are shared in a tribe under majority rule. These tags mimic cultural memes, and are transmitted and conserved. When tags are associated with survival and reproductive success, the tribes show Spencerian cultural evolution.

Similar rules led to combat, division of labor, disease transmission, and other evocative results.

Currently, ABMs typically entail:

1. Many agents, often with differentiated roles and thus diffent rule sets.
2. Rules, which may be complex. Sometimes the rules are heuristic, sometimes one has randomized rules.
3. Learning and adaptation. Agents learn about their environment (including other agents). (Axelrod's Prisoner's Dilemma is an example.)
4. An interaction topology. This defines which agents affect each other—usually this is a model of propinquity, but for auctions it is a star-graph.
5. A non-agent environment. This may include initial conditions, and/or background processes.

Since the 1990s, most simulations use smart agents with multiple roles. Current work is exploring how well agents that implement models for human cognition (e.g., bounded rationality) can reproduce empirical economic behavior.

2. Statistics for Agent-Based Models (ABMs)

Conspicuously absent in the preceding history was statistics. Although statistics has had key interactions with equation-based simulation, it has not been part of the ABM literature.

The main point of this talk is to lay out statistical approaches to ABM simulations, and to encourage the field to study their theoretical properties.

Specifically, we shall discuss paramerization, validation/calibration, and uncertainty assessment. To provide a concrete example for addressing these topics, we shall focus on models for disease transmission.

2.1 Parameterization

Consider an old-fashioned differential equations model for disease spread. The Kermack-McKendrick model assumes:

$$\begin{aligned} \frac{dI}{dt} &= \lambda IS - \gamma I \\ \frac{dS}{dt} &= -\lambda IS \\ \frac{dR}{dt} &= \gamma I \end{aligned}$$

Here $I(t)$ is the number of infected people, $S(t)$ is the number of susceptible people, and R(t) is the number of people who have recovered and are immune. The λ is the infection rate and γ is the recovery rate.

This is a compartmental model with three compartments—infected, susceptible, and recovered. The model does not fit data espeically well, but it represents a standard approach for describing a simple contact process.

It is clear that the intrinsic dimension of the Kermack-McKendric model is 2. Knowing this enables analysts to explore the simulation space by varying these two parameters, λ and γ, and studying how the response (say, duration of the epidemic) depends on these values.

In contrast, many ABM simulations have been developed for this application. Agents walk around a space, infecting each other with a certain probability, and analysts examine the duration.

The ABM simulation is not completely equivalent to the differential equation model—the disease can burn out before the entire population has contracted it. Nonetheless, qualitatively, the two models are extremely similar.

ABM users do not realize that their model is essentially two-dimensional.

The dimensionality of a model is a key property that drives inference and governs complexity. In general, an ABM analysis does not know the intrinsic dimension of the data. It is related to the rule set, but that relationship is usually unclear.

One would like a way to estimate the intrinsic dimension of an ABM. For the disease transmission example, the parameter space is $\mathbb{R}^+ \times \mathbb{R}^+$, but in more complex applications the space will be more complex (often a Cartesian product of intervals and nominal values).

When the intrinsic parameter space is a convex subset of a Euclidean space, then one can probably estimate the dimensionality by principal components regression (PCR). For example, one starts the ABM with many different values for movement and disease transmission, records the duration of the epidemic, and then does PCR to find the number of components required to explain a substantial fraction of the variability in the response.

However, the intrinsic parameter space is probably not usually a convex subset of a Euclidean space. But it may still be possible to get an estimate of the local dimensionality of the ABM.

Run the AMB many times. Let y_i be the output of interest for run i, say the duration of the epidemic. And let $\boldsymbol{x}_i \in \mathbb{R}^p$ be all the tunable parameters in the ABM (e.g., infectiousness, mixing, family sizes, etc.). Then iterate the following steps M times.

1. Select a random point $\boldsymbol{X}^*_m$ in the convex hull of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, for $m = 1, \ldots, M$
2. Find a ball centered at $\boldsymbol{X}^*$ that contains exactly k points (say $k = 4p$).
3. Perform a principal components regression on the k points within the ball.
4. Let c_m be the number of principal components needed to explain a fixed percentage (say 80%) of the variance in the y_i values.

The average of $c_1, \ldots, c_M$ may be a useful estimate of the average local dimension of the ABM model.

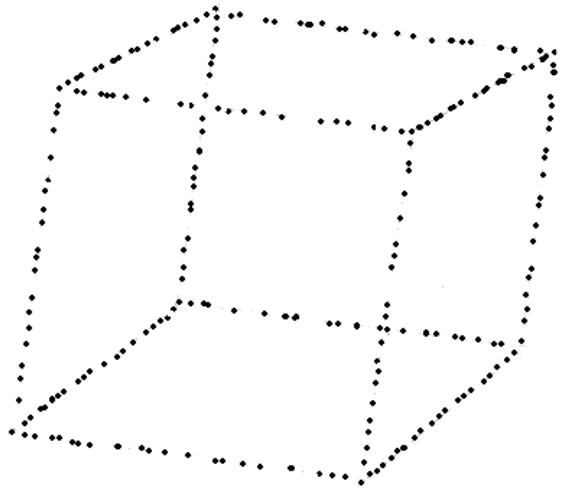
This heuristic approach assumes a locally linear functional relationship for points within the ball. The Taylor series motivates this, but the method will break down for some pathological functions or if the data are too sparse.

To test the approach, Banks and Olszewski (2003; *Statistical Data Mining and Knowledge Discovery*, 529-548, Chapman & Hall) performed a simulation experiment in which random samples were generated from q-cube submanifolds in $\mathbb{R}^p$, and the approach described above was used to estimate q.

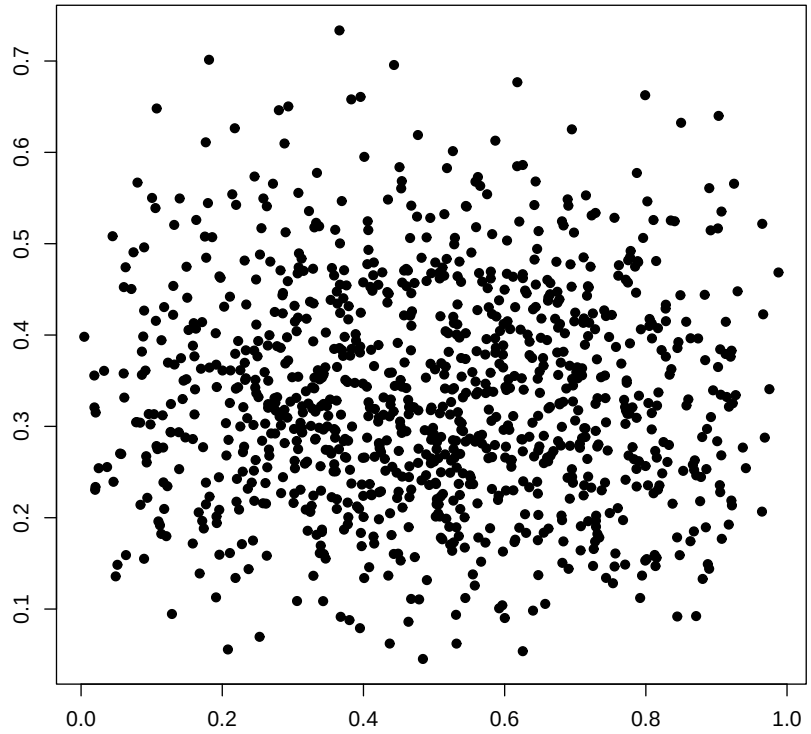
A **q-cube** in $\mathbb{R}^p$ is the q-dimensional boundary of a p-dimensional cube. Thus:

- a 1-cube in $\mathbb{R}^2$ is the perimeter of a square,
- a 2-cube in $\mathbb{R}^3$ are the faces of a cube,
- a 3-cube in $\mathbb{R}^3$ is the entire cube.

The following figure shows a 1-cube in $\mathbb{R}^3$, tilted 10 degrees from the natural axes in each coordinate.



The following figure shows a 1-cube in $\mathbb{R}^{10}$, tilted 10 degrees from the natural axes in each coordinate.



Diaconis and Freedman (1984; *Annals of Statistics*, **12**, 793-815) show that in high-dimensions, nearly all projections look normal.

The simulation study generated $10 * 2^q$ points at random on each of the $2^{p-q} \begin{pmatrix} p \\ q \end{pmatrix}$ sides of a q-cube in $\mathbb{R}^p$. Then iid $N(\mathbf{0}, .25\boldsymbol{I})$ noise was added to each observation and the principal components approach was used to estimate q for all values of q between 1 and p for $p = 1, \ldots, 7$.

The following table shows that the method was reasonably successful in estimating the local dimension. The estimates are biased, since the principal components analysis identified the number of linear combinations needed to explain only 80% of the variance. One should probably do some kind of bias correction to account for the underestimate.

Note: This example does principal components analysis rather than principal components regression, but the concept is straightforward.

Note: It seems unlikely that the map from the ABM rule set to the intrinsic parameter space to the output is everywhere high-dimensional. If it is, then there is probably not much insight to be gained.

q							
7							5.03
6						4.25	4.23
5					3.49	3.55	3.69
4				2.75	2.90	3.05	3.18
3			2.04	2.24	2.37	2.50	2.58
2		1.43	1.58	1.71	1.80	1.83	1.87
1	.80	.88	.92	.96	.95	.95	.98
	p=1	2	3	4	5	6	7

The value of p indicates the apparent dimension, while q is the true dimension of the data. Each entry is an estimate of q, and the largest standard error in the table is .03.

2.2 Model Validation/Calibration

Work in climate forecasting at NCAR and explosion simulation at LANL has led to a new approach to calibrating computer models. This is closely related to validation, and the new theory is pertinent to ABMs.

The calibration approach is useful when one has a complex ABM that takes a long time to run, but there is some real-world experimental data which can be used to tune the model.

The goals are to:

- use the experimental data to improve the calibration parameters (i.e., the rule sets);
- make predictions (with uncertainty) at new input values;
- estimate systematic discrepancies between the ABM and the world.

Suppose that at various settings of the rule-based input values $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ one can observe real world responses $y_1, \ldots, y_n$. Let

$$y_i(\boldsymbol{x}_i) = \psi(\boldsymbol{x}_i) + \epsilon(\boldsymbol{x}_i)$$

where $\psi(\boldsymbol{x}_i)$ denotes the real or expected response of the world and $\epsilon(\boldsymbol{x}_i)$ denotes measurement error or random disturbance.

The observed data are then modeled statistically using the simulator $\eta(\boldsymbol{x}_i, \boldsymbol{\theta})$ at the best calibration value $\boldsymbol{\theta}$ as:

$$y(\boldsymbol{x}_i) = \eta(\boldsymbol{x}_i, \boldsymbol{\theta}) + \delta(\boldsymbol{x}_i) + \epsilon(\boldsymbol{x}_i)$$

where the random term $\delta(\boldsymbol{x}_i)$ accounts for the discrepancy in the simulator and $\boldsymbol{\theta}$ is the best, but unknown, setting of the calibration inputs $\boldsymbol{t}$.

Additionally, one also has supplementary data in the form of simulation results $\eta(\boldsymbol{x}_j^*, \boldsymbol{t}_j^*)$, for $j = 1, \ldots, m$. Typically the ABM code takes a long time to run, so m is small.

The Kennedy-O'Hagan approach scales all inputs to the unit hypercube. Then the Bayesian version uses a Gaussian process to model the unknown function $\eta(\cdot, \cdot)$.

In most of the work so far, the Gaussian process has a constant mean function and a product covariance with power exponential form. (See Higdon et al., JASA 2008.)

$$\mathbf{Cov}[(\boldsymbol{x}, \boldsymbol{t}), (\boldsymbol{x}', \boldsymbol{t}')] = \lambda_\eta^{-1} R((\boldsymbol{x}, \boldsymbol{t}), (\boldsymbol{x}'; \boldsymbol{\rho}_\eta)$$

where λ_η controls the marginal precision of $\eta(\cdot, \cdot)$ and $\boldsymbol{\rho}_\eta$ controls the strength of dependency in each component of $\boldsymbol{x}$ and $\boldsymbol{t}$.

It is often useful to add a little white noise to the covariance model to account for small numerical fluctuations (from, say, adaptive meshing or convergence tolerances).

The formulation of the prior is completed by specifying independent priors for the parameters controlling $\eta(\cdot, \cdot)$: the μ, λ_η, and $\boldsymbol{\rho}_\eta$.

A similar Gaussian process model is used for the discrepancy term $\delta(\boldsymbol{x})$. This has mean zero and covariance function

$$\mathbf{Cov}(\boldsymbol{x}, \boldsymbol{x}') = \lambda_\delta R((\boldsymbol{x}, \boldsymbol{x}'); \boldsymbol{\rho}_\delta).$$

This is a bit athletic, and one might wonder whether the approach is actually buying any reduction in the overall cost of simulation or uncertainty about the computer model. But this structure allows one to do Markov chain Monte Carlo sampling to estimate the posterior distribution.

In particular, one gets posterior distributions for:

- the $\eta(\boldsymbol{x}, \boldsymbol{t})$, which is the hard-to-know implicit function calculated by the ABM;
- the optimal calibration parameter $\boldsymbol{\theta}$;
- the calibrated simulator $\eta(\boldsymbol{x}, \boldsymbol{\theta})$;
- the physical system $\psi(\boldsymbol{x})$; and
- the discrepancy function $\delta(\boldsymbol{x})$.

The latter, of course, is most interesting from the standpoint of model validation. When and where this is large points up missing structure or bad approximations.

In particular, iterated application of this method should permit successive estimation of the discrepancy function. One winds up with an approximation to the ABM in terms of simple functions, and the accuracy of the approximation can be improved to the degree required.

2.3 Uncertainty Assessment

One wants to be able to make probability statements about the output of an ABM. For example, if the ABM predicts that 5,000 people will die of the flu in 2010, it is helpful to have some error bars.

Fortunately, this aspect of ABMs may be fairly straightforward. As in traditional statistical simulations, one puts a subjective distribution over each of the input parameters (in the ABM case, these are the tunable values in the rules). Then one makes multiple runs, and uses the standard deviation of the results to make uncertainty statements.

However, this addresses only part of the uncertainty in an ABM—the portion that relates to the tunable parameters. It does not address model uncertainty.

Model uncertainty is a standard problem in statistical inference. For ABMs, model uncertainty relates to which rules agents should follow, rather than tunable parameters embedded in the rule.

To be clear, in the context of disease, one might have a rule that agents contact a Poisson number of people per day, with the mean of the Poisson being a tunable parameter.

But one might consider adding an additional rule—say that people interact preferentially within a social network, rather than meeting people at random. Model uncertainty pertains to whether or not this rule should be included in the ABM.

The usual fix is to use an **ensemble**, in which there are many different models, and their predictions are weighted according to their predictive accuracy and mutual correlations. Compared to standard applications, ABMs may have an advantage in that increasing complexity is "nested", and overly complex rules will not affect the prediction.

3 Conclusions

- ABMs are an important new tool in simulation; starting in the 1990s, they have become a standard technique.
- Statistical theory for ABMs is almost non-existent. We need to pay attention to this, and we have tools that may improve ABM methodology.
- A key step in a formal ABM theory is a better understanding of the parameterization. Probably one wants a map from $\mathbb{R}^p$ to the input space, which the simulator then maps to the output space. But the first map will be weird.
- A second key step is the development of calibration methods for ABMs—right now, users rely upon face validity, and can miss important structure.
- A third key step is uncertainty expression. All simulations encounter this, but ABM users have been slow to address it.
- Relevant statistical theory has been or is being developed.