

Agent-Based Models: A New Challenge for Statistics

David Banks

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.
- Supply chains, disease spread, genetic algorithms.

Often the localization is geographic, but this is not essential. The auction and genetic algorithm examples have 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.

1.1 The Sugarscape

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). They showed how simple and interpretable rules for agents could simulate realistic demography, anthropology, sociology, and economics.

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

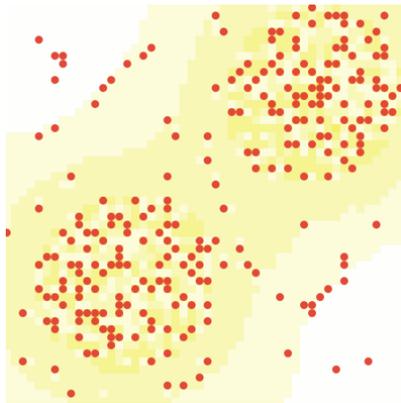
More rules created additional complexity. Epstein and Axtell added sex, and when there was sufficient sugar, the agents would reproduce. This led to age pyramids, carrying capacity, tribal growth with co-movement and fission, and other demographic features.

1. Sugarscape Growback: At each lattice position, sugar grows back at a rate of α per time interval up to the capacity of that position.
2. Agent Movement: Look out as far as vision permits in each of the four lattice directions, north, south, east and west:
 - Considering only unoccupied lattice positions, find the nearest position producing maximum welfare;
 - Move to the new position;
 - Collect all the resources at that location.
3. Agent Mating:
 - Select a neighboring agent at random;
 - If the neighboring agent is of the opposite sex and if both agents are fertile and at least one of the agents has an empty neighboring site then a newborn is produced by crossing over the parents' genetic and cultural characteristics;
 - Repeat for all neighbors.

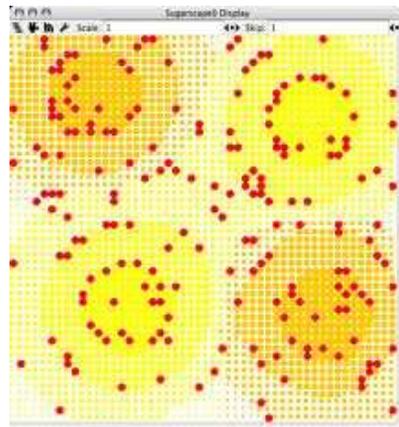
Note that the first rule includes a tunable parameter; there are many such cases in the full list, and this is common in ABMs in general.

Epstein and Axtell added “spice”, a second resource similar to sugar, and simple rules led to barter economies. Then 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.

Additional rules allowed pollution, diffusion of pollution, accumulation of wealth, the evolution of genetic traits, the spread of disease, specialized labor, and combat. In all, 17 rules were sufficient to produce a rich range of social behavior.



Surgarscape



Migration

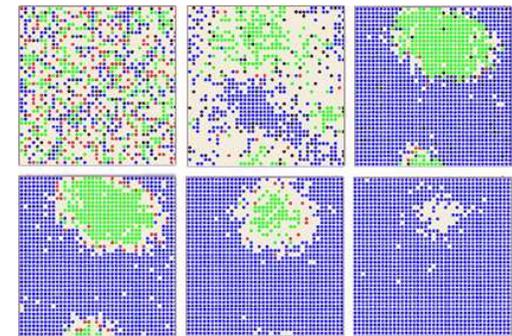


Figure 6.

Conversion

1.2 Kauffman's Model

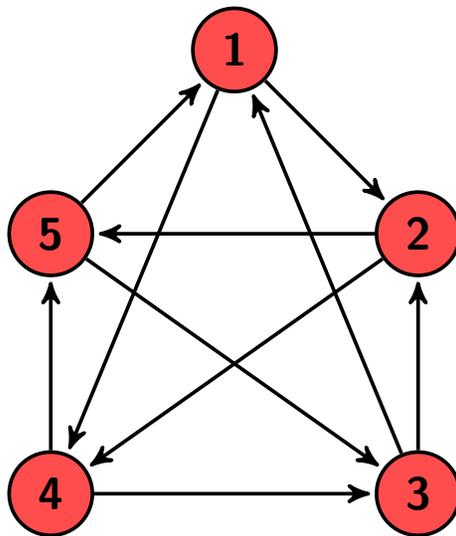
Stanley Kauffman wanted to know how DNA could give rise to the number of different tissue types that are observed in organisms. He performed an ABM experiment described in “Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets”.

In Kauffman's model, each agent is a gene. Each gene is either off or on, 0 or 1, depending upon the inputs it receives from other genes. Genes receive inputs from other genes, calculate a Boolean function, and send the result to other genes.

If a gene receives k inputs, then there are 2^k possible vectors of inputs. And for each given vector of inputs, there are two possible outputs, 0 or 1. So the number of possible functions that map $\{0, 1\}^k \rightarrow \{0, 1\}$ is 2^{2^k} .

input		output	input		output	input		output
0	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1
1	0	0	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1

The first table corresponds to Boolean operator AND, the second is OR, and the last is TAUTOLOGY, one of two degenerate operators that were excluded. Operator names derive from truth tables used in formal logic.



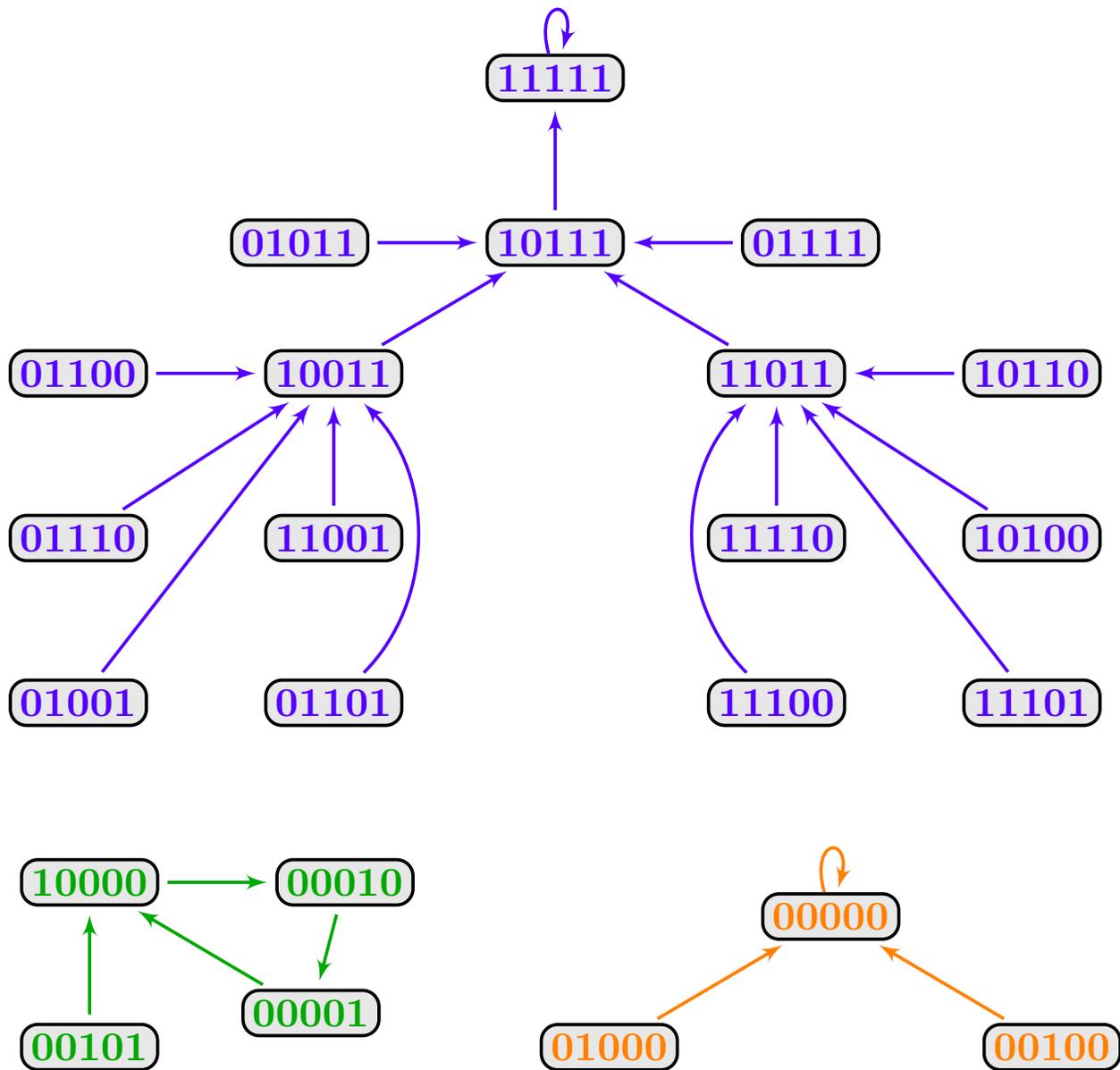
$$B_1(\sigma_5, \sigma_3) = \sigma_5 \text{ OR } \sigma_3$$

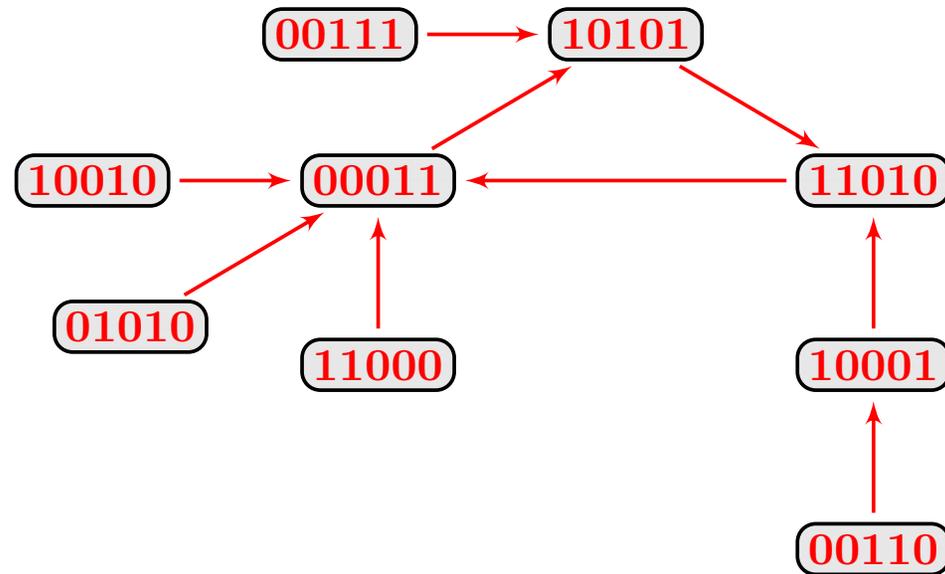
$$B_2(\sigma_1, \sigma_3) = \sigma_1 \text{ AND } \sigma_3$$

$$B_3(\sigma_4, \sigma_5) = \sigma_4 \text{ AND } \sigma_5$$

$$B_4(\sigma_1, \sigma_2) = \sigma_1 \text{ OR } \sigma_2$$

$$B_5(\sigma_2, \sigma_4) = \sigma_2 \text{ OR } \sigma_4$$





These figures show how the randomly connected Boolean network, under various initializations, transitions to different stable behaviors, where each stable behavior is an absorbing state or cycle.

Kauffman used his ABM to simulate repeated initializations and connections, discovering that the number of different stable behaviors is asymptotically equal to \sqrt{n} , where n is the number of nodes. Biologically, each stable solution corresponds to a tissue type. These random networks are robust—if one flips one or two bits at random, the system usually returns to the previous stable behavior.

2. Modern ABMs

Currently, ABMs typically entail:

1. Many agents, often with differentiated roles and thus different 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.

Recent work is exploring how well agents that implement models for human cognition (e.g., bounded rationality) can reproduce empirical economic behavior.

Few statisticians have looked at ABMs. Unlike linear models, in general we don't know how to assess fit, estimate parameters, or make statements about predictive accuracy with ABMs.

Hooten and Wikle (2010) introduced Bayesian hierarchical models into ABM research. Their method applies to some ABM problems, such as spatio-temporal processes with fairly simple structure.

Their application was the spread of rabies in raccoons in Connecticut between 1991 and 1995. On a gridded map representing the townships in the state of Connecticut, they represented the presence or absence of rabies by a binary random variable whose distribution depended upon the states in the neighboring townships at the preceding time period, as well as covariates (which could also vary in time).

Their model allows disease spread to be anisotropic (e.g., along the Connecticut river). It enables statements about the posterior probability of disease in specific townships, using MCMC.

Let $\mathbf{u}_t = (u(1, t), u(2, t), \dots, u(m, t))'$ denote the binary vector showing the presence or absence of rabies at time t for each of the m townships and let $\mathbf{X}_t = (\mathbf{x}(1, t), \mathbf{x}(2, t), \dots, \mathbf{x}(m, t))$ denote a matrix of corresponding covariates, such as population density, adjacency to the Connecticut river, and so forth. Define the neighborhood for township i by \mathcal{N}_i ; this is a set of townships. Then the basic model for the spread of the disease is

$$u_{i,t} \mid \mathbf{u}_{\mathcal{N}_i, t-1} \sim [u_{i,t} \mid h(\mathbf{u}_{\mathcal{N}_i, t-1}, \mathbf{x}_{\mathcal{N}_i, t-1})]$$

where $h(\cdot, \cdot)$ is a very general updating function, the subscript \mathcal{N}_i indicates the townships relevant to the disease spread at township i (i.e., its neighboring townships), and the bracket notation indicates that the presence or absence of rabies is a random variable with parameters that depend on the conditioning within the bracket.

The only substantive difference between this model and a Gaussian state space model is that the random variables need not be Gaussian (which generally precludes closed-form solution, putting this in the realm of ABM simulation).

The main problem with general inference for ABMs is that one does not have a likelihood function. In that situation there are only two ways forward:

- Emulators, which build a tractable model that approximates the ABM (usually some form of Gaussian process);
- Approximate Bayesian Computation (ABC), in which parameter values are simulated from the prior, the ABM is evaluated at those values, and if the outcome is too far from observed outcomes, that parameter value is discarded, downweighting its value in the posterior.

Bijak, Hilton, Silverman and Cao (2013) apply an emulator to an ABM to the demographic study of marriage.

Farah, Birrell, Conti and De Angelis (2014) examined emulators in the context of a model for the spread of H1N1 influenza.

Heard (2014) compared emulators and ABC in the context of HIV spread and simulated drug markets. He found that emulators were generally more tractable.

3. Emulators

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 $\mathbf{x}_1, \dots, \mathbf{x}_n$ one can observe real world responses y_1, \dots, y_n . Let

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

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

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

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

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

Additionally, one also has supplementary data in the form of emulator results $\eta(\mathbf{x}_j^*, \mathbf{t}_j^*)$, for $j = 1, \dots, m$. 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., 2008.)

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

where λ_η controls the marginal precision of $\eta(\cdot, \cdot)$ and $\boldsymbol{\rho}_\eta$ controls the strength of dependency in each component of \mathbf{x} and \mathbf{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 $\delta(\mathbf{x})$. This has mean zero and covariance function

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

As before, there are some important technical details regarding the the specification of the form of the covariance function.

This is all a bit athletic, and one might wonder whether the approach is actually buying any reduction in the overall cost of emulation or uncertainty about the computer model. But this structure enables one to do Markov chain Monte Carlo sampling to estimate the posterior distributions for critical ABM outputs.

Specifically, one gets posterior distributions for:

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

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.

Iterated application of this method permits 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 often be improved to the degree required by elaborating the Gaussian process model.

4. Model Validation

The **validation problem** is to determine whether an ABM is faithful to reality. ABMs make many different kinds of assumptions and approximations, and often one has little data from the real world.

There are five main approaches to validation:

- Physics-based modeling
- Intelligent design
- Face-validity
- Comparison to another model
- Comparison to the world

Regrettably, none of these is fully satisfactory.

Physics-based ABMs are commonly used in the hard sciences. People build a simulation that incorporates all of the physical laws and interactions that they can think of, and then feel confident that their model is faithful to reality.

This can work on smaller problems where all mechanisms are fully understood. Examples where it is arguably successful include n -body problems and the Ising model. But it tends to break down as the stochasticity of the problem increases. And the micro-level simulations can take a long time to run.

Intelligent design is the most common way to validate an ABM. Experts think through the simulation carefully, building in or approximating all the effects they think are substantial. Then they hope that they've been smart enough and wait to see if their results work well enough. Intelligent design can handle more complicated situations than the physics-based models, but there is no test for statistical validity.

Since the thought-process is similar to software design, the results are probably about as buggy as code.

Face-validity is the first case with a true validation protocol. The designer tests the ABM by using selected inputs to explore the output behavior. Often the selected inputs are chosen according to a statistical design, such as a Latin hypercube. Alternatively, one can try to select values for the inputs that correspond to expected behaviors.

Face-validation fails when the parameter space is large and there are many interactions. But, to varying degrees, it is used for systems such as EPISIMS and the battlefield simulations produced at DMSO.

Model comparison is not often done often, but it has the potential to be a powerful tool for validation. The advantages are that one can better explore the full range of model behavior.

One issue is that it is hard to decide when an ABM is nearly equivalent to another model, or perhaps is a subset of the other. And it can be especially important if one of the simulations runs very much faster than the other; for example, one can use the slow one that is possibly more accurate as an anchor for calibrating the fast one.

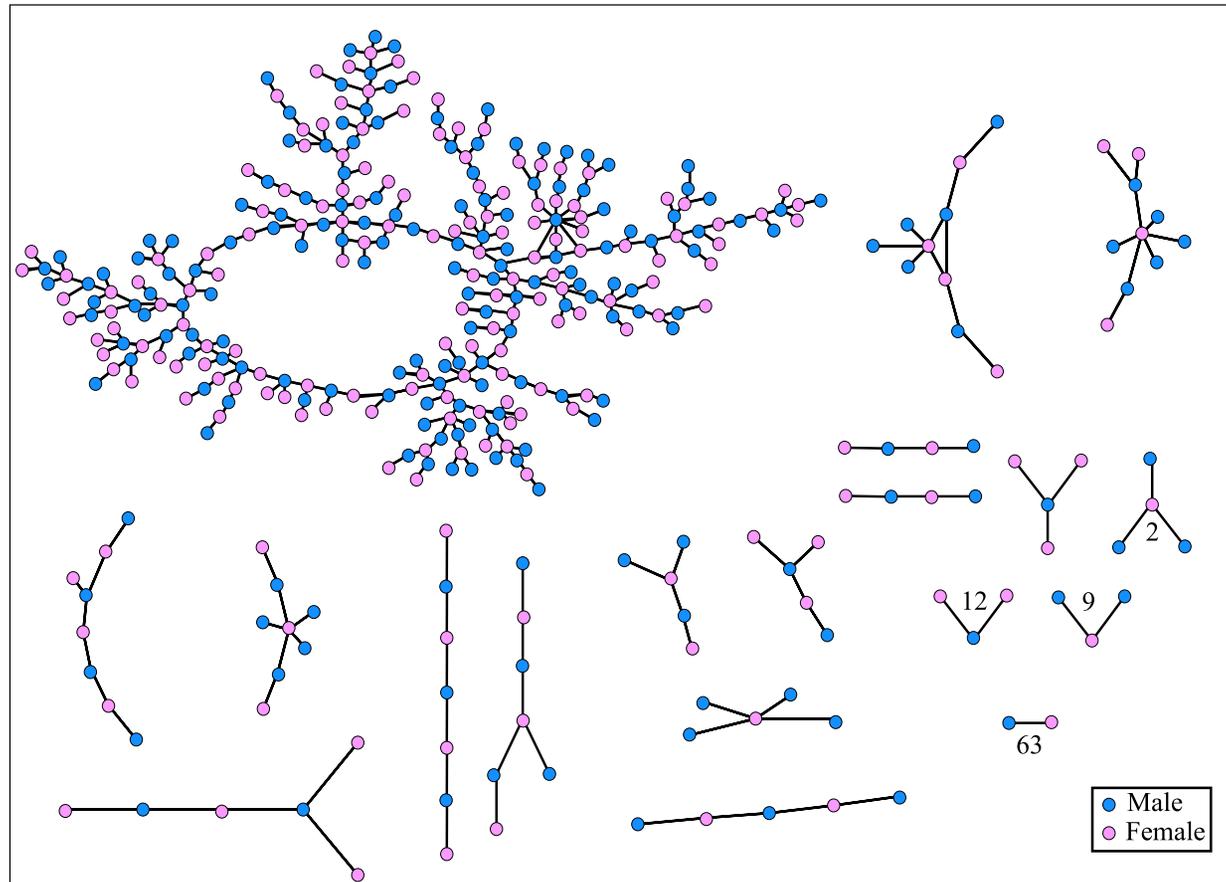
For some situations, such as weather forecasting, one can compare ABM predictions to what was later observed. This is the strongest form of validation, but it is still inadequate. One does not have confidence in the fidelity of the ABM in regimes that have not been previously observed, but this is often the context of the greatest interest.

Also, when there is randomness, it is difficult to quantify the strength of support obtained by correct or incorrect predictions.

An ABM is model just like the linear model, but we do not know how to:

- assess goodness-of-fit,
- estimate the parameters in an ABM, or
- make quantified statements of uncertainty about predictions.

As an example of the importance of ABM validation, consider a study of sexual relationships among 832 students at “Jefferson High” by Bearman, Moody and Stovel (1995).



The Jefferson High data are aggregated over six months, so the STD threat is slightly overemphasized in the graph. The researchers found that there were strong tendencies for racial and smoking homophily, and for gender heterophily.

The researchers used an ABM to generate random networks that had the same degree counts, the same racial/smoking/gender cross-links, and the simulated networks looked very different (more “clumpy”) than the real one.

Then they added an extra condition, to exclude four-cycles. That is, if Bob and Carol are together, and Ted and Alice are together, but later Bob and Alice get together, the new condition says that Ted and Carol cannot get together. (In high school, there is no status gain when two dumpees hook up.)

Simulations with this new constraint looked like the real network.

5. Parameterization

Parameterizing ABMs is tricky. Often, people build in too much detail, leading to unwieldy programs and long run times.

For example, the old Kermack-McKendrick model for disease spread is

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

where $I(t)$ 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 real data 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 and size of the epidemic.

ABMs for disease spread hope to capture effects that compartmental models cannot, such as social distancing and inhomogeneous mixing, and to evaluate the impact of public health interventions, such as school closures or quarantines. But one still needs to estimate the intrinsic dimension of the ABM—if it is very high, then forecasts will be unreliable.

Often the behavior of interest is globally high-dimensional but locally low-dimensional.

5.1 Estimating Intrinsic Dimensionality

The dimension 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 SIR example, the parameter space is $\mathbb{R}^+ \times \mathbb{R}^+$, but in ABMs the space will be more complex (often a Cartesian product of intervals and nominal values), corresponding to the rules that the agents follow.

When the intrinsic parameter space is a convex subset of a Euclidean space, then one can estimate the dimension by principal components regression (PCR). One starts the ABM at different parameter values, records the output, and then performs PCR to find the number of principal components needed to explain a substantial fraction of the variability in the response (say, 80%).

But the intrinsic parameter space for an ABM is usually not a convex subset of a Euclidean space. But one can still estimate of the local dimension 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 $\mathbf{x}_i \in \mathbb{R}^p$ 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 \mathbf{X}_m^* in the convex hull of $\mathbf{x}_1, \dots, \mathbf{x}_n$, for $m = 1, \dots, M$
2. Find a ball centered at \mathbf{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, \dots, c_M provides an estimate of the average local dimension of the ABM model.

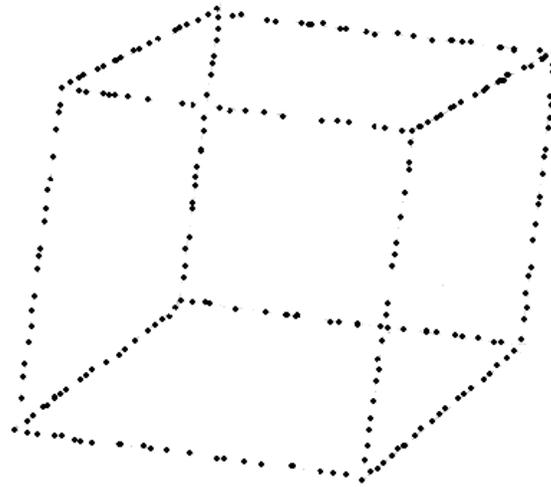
This approach assumes a locally linear functional relationship for points within the ball. The Taylor series motivates this, but the method will break down for some pathological functions or if the data are too sparse.

To test this method for estimating average local dimension, Banks and Olszewski (2003) 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 .

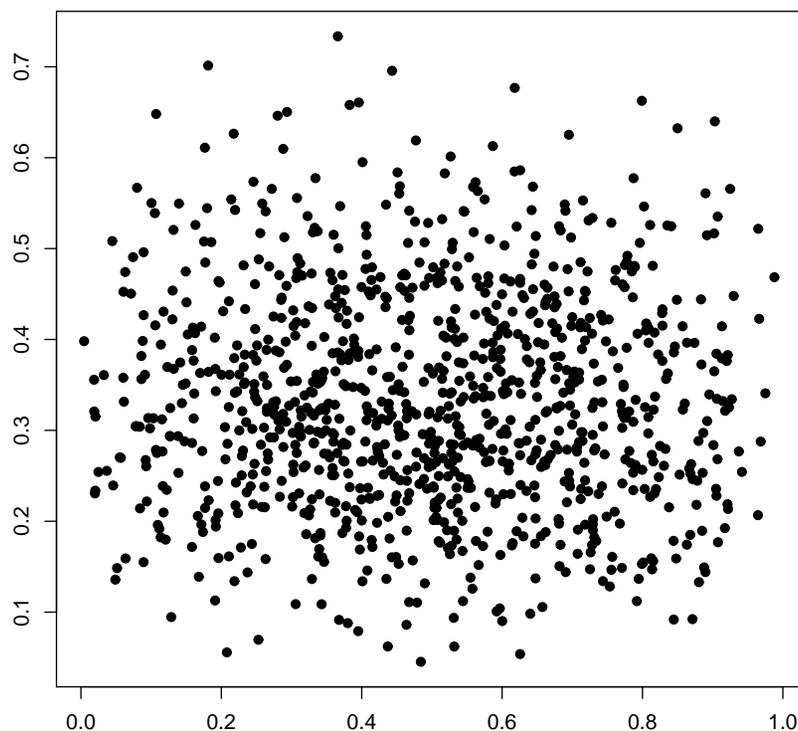
A **q -cube** in \mathbb{R}^p is the q -dimensional boundary of a p -dimensional cube. Thus:

- 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) show that in high-dimensions, nearly all projections look normal.

The simulation study generated $10 * 2^q$ points at random on each of the $2^{p-q} \binom{p}{q}$ sides of a q -cube in \mathbb{R}^p . Then iid $N(\mathbf{0}, .25\mathbf{I})$ 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 $p = 1, \dots, 7$.

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 could do bias correction to account for the underestimate.

Note: This example does principal components analysis rather than principal components regression, but the extension 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 were, 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.

5.2 Model Equivalence

Suppose one has two ABMs for the same process. One may want to decide whether the models are equivalent, equivalent in mean, equivalent modulo monotonicity, or equivalent in distribution.

This is a hard problem. Heard (2014) proves that for models that correspond to continuous mappings, two models are topologically equivalent if and only if the atoms of their critical values are isomorphic. And topological equivalence implies that there exists a calibration function and a tuning function that makes the models exactly equivalent. But verifying this condition in any realistic application is not practical.

Nonetheless, it is important to know when two models are essentially the same, or when two models are fundamentally different. With complex ABMs, this is often not obvious.

6. 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.