Statistical analysis of agent-based models

Dr Jason Hilton and Prof. Jakub Bijak, University of Southampton

Hello. In this video, we would like to introduce the basic concepts related to the design of experiments, used to help us understand the behaviour of complex computational models, such as agent-based models. We will look at the relations between inputs and outputs of models, types of experimental design, and methods of analysing the results of simulation experiments. We will look particularly closely at the statistical emulators – or metamodels – and methods for uncertainty and sensitivity analysis.

So, what is an experiment? What do we learn by experimenting that we would not be able to learn otherwise? Why is an appropriate planning of experiments so important?

In our context, we will interpret an experiment as a process of measuring a response of some output variable to changes in a set of inputs. A special case of an experiment is a computer-based one, which is based on a mathematical description, implemented on a computer by using numerical methods and algorithms. By nature, computer experiments are replicable and can have higher speed and lower cost than for example physical, chemical, or biological experiments. This makes them well suited to studying large-scale complex social systems. To do that, computer experiments typically rely on deterministic or stochastic simulations, which ideally need to be transparent and well documented.

Computer experiments for studying social research questions also have different ethical considerations than other types of experiments, especially those, which would require human participation. For instance, if we were to analyse optimal ways of evacuating people facing immediate danger, such as fire or flood, the process cannot involve live experiments in real dangerous conditions. Here, computer experiments can help by simulating the situation and the evacuation process. This is one example, where agent-based models coupled with computer experiments can help.

How to do that in practice? To make the most of computer experiments, they need to be planned and designed in advance. To ensure that we gain maximum information, we need to know at which parameter values and with which settings the models need to be run. This is the domain of a branch of statistics called *design of experiments*. Its key aims are to help understand the relations between model inputs and outputs, and to maximise information or to minimise the error of experimentation.

The literature on the design of experiments has its specific terminology. On the next few slides, we would like to introduce some of the key terms. Their definitions are mostly based on the online compendium on Managing Uncertainty in Complex Models.

First, a simulation model, or a simulator, such as an agent-based model, is a computer program that represents some real system – in our case, a social one. The simulator has some inputs and produces an output, which is the variable of interest to the modeller. The input, or factor, is a variable that can be controlled in the model. The inputs can include model parameters, or other aspects of model specification. At the same time, an output, or response, represents one of the key features of the real system that is being modelled. The relationship between inputs and outputs is that an individual run of a model, based on a set of inputs, produces – deterministically or stochastically – an output.

Second, the model is aligned with the reality – or rather, with observed data – in the process called *calibration*, which aims to make model outputs as close to the observations as possible. It does so by systematically changing the inputs. If an input has a best value that minimises the difference between the output and the observations, such input is called a calibration parameter. Whatever is left in terms of a difference between the observed reality and the output of a calibrated model, is called a model discrepancy, or inadequacy.

This slide illustrates the concepts of calibration parameters and model discrepancy. Let's assume that the real world process is a sine function shown as a solid dark line, and the observations are the black points. We don't know the model, but suspect that it may be a sixth-order polynomial. We fit such a polynomial by using ordinary least squares, and the result is as shown with a grey dashed line. The coefficients of this polynomial are here calibration parameters, and the large gap that remains between the model and reality is the model discrepancy.

To help with the analysis, we need to introduce a few more terms. Hence, a meta-model, also known as an emulator, or a surrogate, is a statistical or mathematical model of the simulation model. By 'design' or 'design space', we will understand a selection of input points, at which the model will be run. The part of the design space which will be used to calibrate a model or to build a meta-model is called a 'training sample', and includes inputs, as well as their corresponding outputs.

In simulation models, such as agent-based models, the relationships between inputs and outputs are not obvious. There are many sources of uncertainty, some of which are related to the reality and our imperfect knowledge about it, while other to the different elements of the model construction, including the parameters and the computer code.

Using the tools of statistical experimental design to analyse the results of agent-based models is based on the observation that the results of agent-based models, no matter how opaque, are indeed computer-based experiments. If we run our model at different parameter values and with different settings – so if we *experiment* by executing the model on a computer many times, we learn about the behaviour of the model. This is especially important given the sometimes very complex, non-transparent and analytically intractable nature of many computational simulations.

At the same time, we need to be transparent about the uncertainty of complex models. This is the domain of uncertainty quantification, or UQ. UQ is a research area which involves learning about model uncertainty, sensitivity of the output to different inputs, and finding out about model parameters by calibrating the model to available data. Bayesian methods offer natural tools for UQ applications.

In the second part of this video, my colleague, Dr Jason Hilton, will say more about the different methods that are used for designing and analysing experiments involving complex agent-based simulation models – and of course their uncertainty.

Hello. As you've just heard, we have powerful tools that enable us to better understand both simulation models and they way in which they relate to the real-world social process they represent. Properly accounting for uncertainty is vital, as without doing so we are running the risk of drawing incorrect conclusions about our model.

In the rest of this lecture, I will describe the basic principles of design and emulation, before moving on to uncertainty quantification and calibration techinques.

One key insight is to think about a computer simulation as a function, represented here by f(x), that transforms input values to output values. We can approximate this function using a statistical model – an emulator. It provides a stand-in for the simulation that is simple and transparent, and which can estimate outputs of the simulation almost instantaenously for any combination of input points.

In attempting to construct such a model we need to be able to answer two questions. Firstly, at what design points should we run our simulation in order to create the *training sample* for our meta-model. And secondly, what *type* of meta-model should we use to emulate the behaviour of the simulation.

These questions have important practical implicatoins. If your simulation takes a long time to run, poorly chosen experimental designs may mean that computations take too long before results can be obtained. With inappropriate meta-models, a poor approximation to your simulation may be obtained, leading to faulty conclusions. In fact, the answers to these questions are strongly intertwined: both choice of design and the choice of meta model rely on the types of assumptions we are willing to make about how the simulation in question transforms inputs to outputs.

For simple simulations where the relationships between simulation inputs and outputs are expected to be linear, or perhaps involve simple two-way interactions or quadratic effects, a least-squares regression model with low-order polynomial terms will be enough. These polynomial regression models can be fitted with *factorial designs*, which involve defining two or three levels for each input dimension, and running the simulation at every combination of these levels.

However, Agent-Based Models involve the mutliple interaction of many random decision-makers, and as such may exhibit *complex* behaviour. This may result in highly non-linear relationships between inputs and outputs. As a result, more flexible metamodels may required, combined with experimental designs which allow for more complicated input-output relationships to be detected.

Gaussian Process emulators are one such class of meta models. These are non-parametric Bayesian statistical models that make much less restrictive assumptions about the simulation. More specifically, they assume that the relationship outputs are smooth functions of inputs values, and that input points that are closer together will lead to similar outputs.

These assumptions are codified in a *prior distribution,* samples from which are displayed in the figure on the slide. This describes the sorts of input-output relationships that we believe might be possible *before having run the simulation.* Once the meta-model has been updated with training data, a *posterior distribution* can be obtained that describes our prediction for the simulation output for any combination of input values, based on what we have learnt about the simulation from the training data. We also obtain a measure of how certain we are about the value of the simulation at input values we have yet to observe. Such a posterior distribution, updated to account for the outputs observed at four training design points (displayed as red dots), is displayed in the bottom figure of the slide.

Factorial designs may not provide sufficient resolution to allow discovering complex relationships between inputs and ouputs. The number of simulation runs needed becomes very large when more inputs are used. A *latin hypercube design* divides each input axis into a number of sections, and ensures that there is a least one design point in each section. Latin hypercubes provide a good designs for experiments with ABM because they can be made to be *space filling,* so that all areas of parameter space are explored, and because they don't require so many points when many simulation inputs are to be investigated.

The combination of latin hypercubes and Gaussian process emulators are good default choices for the analysis of Agent-Based Models, providing for quick approximations of the simulation output for any inputs values. They also allow for an appropriate description of the uncertainty associated with these approximations and resulting from other sources.

Uncertainty quantification describes a set of tools aimed at providing rigourous accounting for what we don't know about a simulation and its relationship with the reality it is supposed to represent. Bayesian statistics, which interprets probability as representing degrees of belief, is a natural framework within which to conduct such quantification exercises.

Uncertainty relating to computational experiments comes from a number of sources. The first one relates to the simulation itself, and how much we know about its behaviour. This includes the extent to which pseudo randomness in the code translates to randomness in the output, and our lack of knowledge about the strength with which outputs respond to inputs.

Another source of uncertainty corresponds to how the simulation relates to reality. Firstly, we may not know the true value of our simulation inputs – that is, the value which holds in reality. Secondly, the simulation model we are analysing may be an imperfect representation of reality and may be wrong in some key respect in how it represents the real world process. This is the model descrepancy that Jakub described earlier. Finally, the observed values of the real world may rely on imperfect measurement tools, such as surveys subject to sampling errors or measurement instruments subject to bias.

Gaussian process meta-models can be analysed in order to quantify some of these sources of uncertainty. Uncertainty Analysis is one technique for understanding how our uncertainty about true input values results in uncertainty in our ability of predicting the system outputs using our simulation. The extent of our uncertainty about inputs may be estimated using variability in our observations of these values, or constructed based on reasonable prior information.

Sensitivity analysis goes one step further, and describes how sensitive a simulation output is to changes in inputs. In general, this can be done locally – that is, for specific values of the simulation inputs – or globally, in which case we are interested in the average or total sensitivity across all inputs values. Global sensitivity analyses partition the total output variance due to each changes in each input, in a similar way to more traditional ANOVA techniques.

Meta-models can also aid with the problem of calibrating a simulation; that is, of finding values of unknown simulation inputs that result in simulation outputs that match the values of these quantities that are observed in reality. One way of doing this is by optimising; by defining some measure of distance or error that compares simulated outputs and observations, and minimising this measure by adjusting inputs. The single input point that results from this process is an optimal input value.

However, given the list of uncertainties we discussed earlier, we are unlikely to know enough about our system to be confident that such an optimal value is the 'true' one. Relying on such an optimal value may lead us to draw incorrect conclusions from our simulations. Probabilistic calibration methods instead aim to account for all sources of uncertainty, and identify a set of input values that could have generated the values we observe, protecting us against over confidence.

Gaussian process emulators are not the only choice of meta-model. However, they do provide helpful tools for the analysis of agent-based simulations, and for understanding the various sources of uncertainty about both a simulation and how it relates to reality. We must be careful, however, not to think that meta-modelling will solve all our problems for us. A calibrated simulation is not a guarantee of a correct simulation, because there may be infinitely many simulation specifications that can approximate real-world observed values. Only some such simulations provide reasonable approximations of the real data-generating process. A robust process of validation, verification and grounding in data and theory must be in place to have confidence in conclusions drawn from agent-based simulations.

Many thanks for listening.