Probabilistic Spatial Agent-Based Models (for Social Simulations)

Archer L*[1]

[1]Leeds Institute of Data Analytics, University of Leeds, Leeds, LS2 9JT, UK
January 21, 2019

KEYWORDS: Probabilistic Programming, Agent-Based Modelling, Data Assimilation

Social simulations are naturally complex and difficult to model, involving a great deal of stochasticity from diverse sources. Individual-based modelling techniques, such as Agent-Based Models (ABM), are ideally suited to modelling the dynamics of these systems, with examples in crime, human migration, consumer behaviour, and disaster management. ABMs are unique in that the model is specified by rules applied to individual agents, and so models lack the statistical structures from which probability distributions can be determined analytically. However, high resolution models of human systems must account for considerable uncertainty, ranging from uncertainty within data to uncertainty that stems from human decisions. Researchers have attempted to account for this uncertainty by running large numbers of simulations with stochastic agent behaviours to explore the distribution of possible outcomes. This is computationally expensive, and only allows for an approximation of the distribution.

Due to this uncertainty, ABMs of social systems tend to diverge from reality over time. This has led to recent attempts to carry out Dynamic Data Assimilation (DDA) on agent-based models, using techniques such as reinforcement learning, particle filtering, and the Ensemble Kalman Filter. DDA has been thoroughly studied and applied in the field of meteorology, where models are governed by a dynamical system of equations. However, interactions between stochastic agents often produce nonlinear effects at the aggregate level, introducing new challenges when using some of the more sophisticated methods from meteorology.

One way to capture this uncertainty is a probabilistic approach. This involves using Bayesian probability theory to express all forms of uncertainty directly, meaning variables in the model are represented not as a single value but as a distribution over the range of possible values. These probabilistic constructs can then fully represent how uncertainty propagates through model iterations, and allows for computationally efficient inference. Importantly, recent advances in probabilistic programming have enabled probabilistic models to be created that are much more flexible and general than those that are based on traditional graphical frameworks. For example, probabilistic libraries allow common programming operations such as recursion and control flow statements which cannot be represented in a finite graph. A good example of this is Edward (Tran *et al.*, 2017), a Turing complete probabilistic programming language that is integrated into TensorFlow.

However, whilst a probabilistic ABM would be a large part of the solution to capturing uncertainty, and allow easier application of sophisticated DDA algorithms, it would not be without its own significant challenges. For example, in a spatial model an agent's position would be represented as an (x, y) coordinate, whereas in a probabilistic model the position would be represented as a distribution over all possible locations that the agent might be at that moment. This means a complete re-appraisal for how the models fundamentally operate; how do we accurately represent an agent at a single point in 2+ dimensional space without losing information on the distribution? Also, as interaction is such a key part of these models, how can we determine when two agents are close enough to interact if their locations are uncertain? In effect, the probabilistic approach forces a fundamental rethink as to how agent-based models are created and how agent interactions are formalised.

* L.archer@leeds.ac.uk

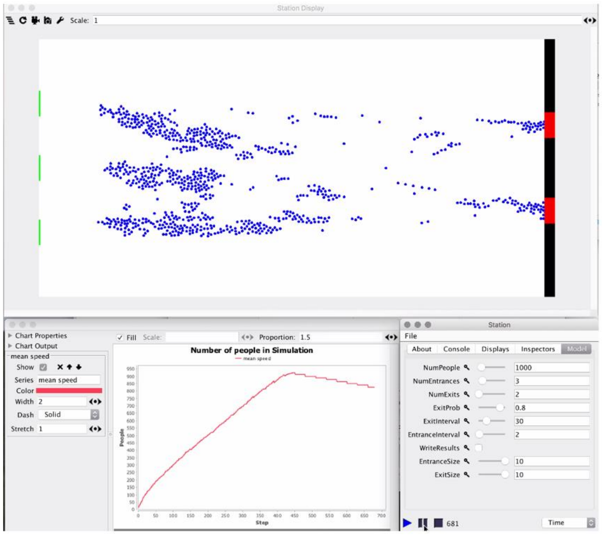

Figure 1. The simple crowding model.

This presentation will discuss ongoing work on probabilistic spatial agent-based modelling. Part of the work is attempting to convert an existing spatial agent-model of crowding (see Figure 1) into a probabilistic model, using the probabilistic programming library *Keanu*, under development by Improbable in London. The model is built using the MASON multi-agent simulation toolkit. Whilst enabling fast and efficient creation of ABMs, MASON is not designed to allow the introduction of probabilistic elements. The interface between Keanu and MASON has provided some significant technical challenges.