State Estimation and Data Assimilation for an Agent-Based Model using a Probabilistic Framework *

Nick Malleson[1,3][0000−0002−6977−0615], Luke Archer[3], Minh Kieu[1,3][0000−0001−7798−6195], Jonathan A. Ward[2][0000−0003−3726−9217], Alison Heppenstall[1,3][0000−0002−0663−3437], and Christoforos Anagnostopoulos[4]

[1] School of Geography, University of Leeds, LS2 9JT, UK
http://geog.leeds.ac.uk/
[2] School of Mathematics, University of Leeds, LS2 9JT, UK
http://maths.leeds.ac.uk
[3] Leeds Institute for Data Analytics (LIDA), University of Leeds, LS2 9JT, UK
http://lida.leeds.ac.uk
[4] Improbable, 10 Bishops Square, London, E1 6EG
http://www.improbable.io

Abstract. This paper outlines a framework for state estimation and data assimilation in agent-based models, using a Bayesian probabilistic framework. It presents initial attempts to begin to quantify the type and volume of data that would be required to realistically model a crowd of people in real time. It contributes to the challenge set out in the workshop theme in two ways. Firstly, by developing methods that support "trusted models that can be used by industry and governments to enhance decision-making, and that can incorporate real (and real-time) data sets in a meaningful way". Secondly, regarding the 'humans and devices' theme specifically, by beginning to estimate the volume of sensors that would be required in order to successfully model crowds in real time. Without a reliable means of incorporating real-time data into urban models, their use will continue to be limited to scenario evaluation based on historical data rather than providing the most likely estimates of the *current* state of urban systems as well as short-term forecasts.

Keywords: Agent-based modelling · Probabilistic programming · Uncertainty · Data assimilation · State estimation · Bayesian inference

* This work was supported by a European Research Council (ERC) Starting Grant [number 757455], a UK Economic and Social Research Council (ESRC) Future Research Leaders grant [number ES/L009900/1], an ESRC-Alan Turing Fellowship [ES/R007918/1] and through an internship funded by Improbable (https://improbable.io/).

1 Introduction and Objectives

Calibration – the process of finding optimal values for a model's parameters – is reasonably well studied in the field of agent-based modelling. Two aligned topics that are much less well studied, however, are those of *state estimation* and *data assimilation*. State estimation refers to the practice of estimating the *true* state of a system. Although their complexity precludes the true state of human systems ever being known precisely, it can be estimated by combining a model of the system with observations (data). The practice of estimating a system's *current* (i.e. real-time) state using a model and some data is often termed data assimilation[1]. Data assimilation has, and continues to be, extremely well studied in fields such as meteorology [4] where it has been credited as being part of the reason that weather forecasts have improved so substantially in recent years [1].

Applications of data assimilation in agent-based modelling are scarce, however; only a handful of attempts have been published [5–7]. The aim of this research is to contribute to this emerging field by performing data assimilation on a simple agent-based model of a hypothetical crowd. The work builds on previous work that was originally presented at ABMUS 2018 [5]. A series of 'identical twin' experiments are planned, whereby the agent-based model first generates 'pseudo-truth' data that reflect the 'true' state of the system (in the real world such data are never available) and then the model is re-run in a data assimilation framework that attempts to replicate the truth data. Importantly, by varying the amount of information about the true system state that is provided to the data assimilation algorithm it is possible to estimate the amount of information that might be required were a real crowd of people to be simulated. Here it will be assumed that some individuals in the hypothetical data are tracked, so their spatio-temporal locations are known, but in future work the framework will be extended to aggregate data as well (i.e. counts of people rather than individual traces). As discussed shortly, the paper also makes use of a relatively novel computational approach: probabilistic programming. Hence the contributions of this paper are two-fold, it:

1. provides a framework to estimate the volume of data are necessary to simulate a crowd in real-time;
2. explores the value the probabilistic approach to modelling as a means of performing data assimilation on an agent-based model.

Experimental results are not yet available, but are under preparation and will be available for the workshop.

2 An Example Agent-Based Model: *StationSim*

[1] It is worth noting that although data assimilation techniques can be used to adjust model parameters in response to new data, here the technique is applied solely to the task of state estimation.

The hypothetical model used here is a simple system of pedestrians, each of whom move from an entrance towards one of two exits. The model environment is illustrated in Figure 1. It has been designed to be as simple as possible and yet meet two criteria:

1. be loosely representative of a spatial human system (in this case the model represents an abstract train station, where passengers arrive on a train and need to cross the station concourse to leave through the exits);
2. exhibit emergence (in this case emergence takes the forms of *crowding* that occurs as a result of the random distributions of agents who all have different maximum walking speeds).

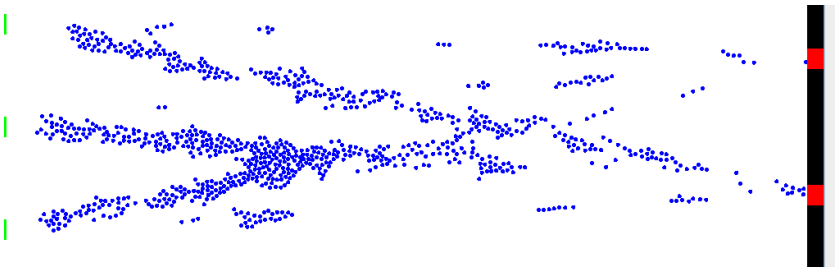

Fig. 1: A snapshot of the simple, hypothetical model that is used here. Agents move from left to right.

The model is Markovian, such that it can produce output from a set of inputs (here termed the 'state vector') without any other information. The *state vector*, at a time t, contains all of the agent ($i = \{0, 1, \ldots, N\}$) variables ($\overrightarrow{v_i}$) but not any model parameters. Therefore

$$S_t = \left[\overrightarrow{v_0}\ \overrightarrow{v_1} \ldots \overrightarrow{v_N} \right] \tag{1}$$

where $\overrightarrow{v_i}$ represents of the x and y coordinates[2] of the each agent (i) at time t:

$$\overrightarrow{v_{i,t}} = \left[x_{i,t}\ y_{i,t} \right] \tag{2}$$

There are numerous model and agent parameters – such as the destination exit and maximum speed for each agent – but these are kept constant throughout the experiments and so need not be included in the state vector. It would be possible to include these parameters in the state vector and estimate them along with the variables, but this will be tested in future work.

[2] Note that the *current* speed of an agent need not be included in the state vector because it can be calculated from an agent's maximum speed and the positions of the other agents.

3 Modelling Framework

Data assimilation is a Bayesian approach. As Figure 2 illustrates, the prior estimate of the state of the (pseudo[3]) real system is generated by running the agent-based model forward from the last point in time that historical observational data were available ($t-1$ in Figure 2). The prior provides uncertain estimates of the current system state, represented in the form of a distribution over the state vector (S). At time t (i.e. 'now'), it is hypothesised that new data have arrived and these need to be assimilated into the model. At present, the locations of all agents in the pseudo-truth are observed, which is analogous to every person in the station having their positions tracked. Although some noise is added to the observations, this is still more data than would usually be available in the real world and later experiments (discussed shortly) will reduce the amount of information provided.

To incorporate the new observations a Bayesian network (a probabilistic graphical model) is constructed that represents the relationships between the different elements of the state vector (the x and y coordinates for all agents) and the station model itself. In effect, each element in the state vector becomes a vertex in the graph, along with the station model.

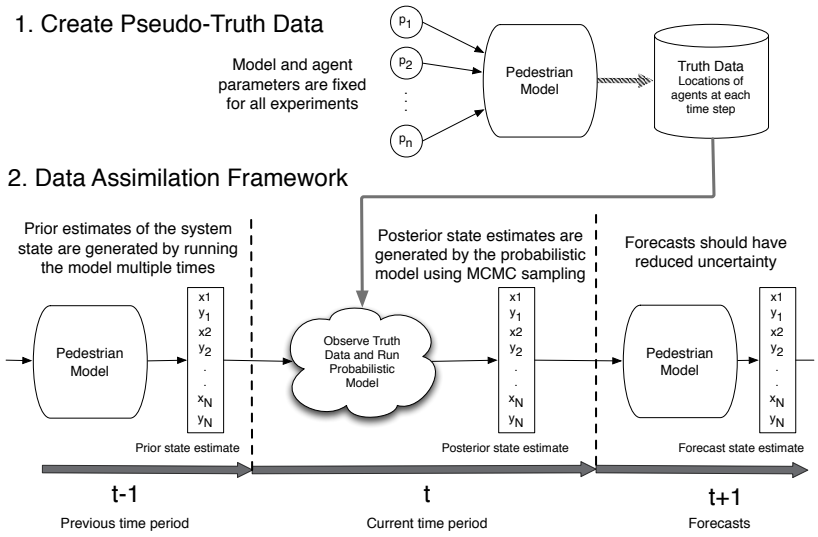

Fig. 2: The framework for conducting state estimation and data assimilation using the identical twin experiment and a probabilistic model. This builds on the initial work outlined in [5].

[3] Recall that, in this case, the 'real' system is actually a single run of the model as per the 'digital twin' experimental framework.

Then, through Markov chain Monte Carlo (MCMC) sampling, posterior estimates of the state vector are produced. Hence the posterior is the new estimate of the system state that is obtained once the most recent observations from the (pseudo) real system have been incorporated. This is useful in itself as it potentially provides a more up-to-date estimate of the current state of the system. In addition, these posterior estimates can then be re-inserted as the new priors to run new model forecasts (i.e. $t + 1$ in Figure 2). As the new forecasts take the most recent available data into account they should be more accurate than forecasts made from earlier, out-of-date data. Here, the whole process is repeated a number of times to reflect the real practice of data assimilation that might take place were a model really attempting to simulate a system in real time.

4 Probabilistic Programming and the Bayesian network

This framework lends itself extremely well to the relatively new approach of 'probabilistic programming' [2, 3]. This is a structure for writing computer programs whereby variables represent probability distributions rather than single values. This approach is therefore ideally suited to Bayesian modelling. Here, a new probabilistic programming library, *keanu*[4], is used. Keanu provides the functionality required to create a Bayesian network and perform MCMC sampling to generate the posterior estimates. Importantly, the Bayesian network is agnostic to the type of information that the nodes in the graph represent. At present, the nodes represent the individual elements of the state vector (i.e. individual agent locations) and observations from the pseudo-truth model map directly onto these nodes. As mentioned above, this is analogous to tracking each individual. However, the probabilistic model can produce a posterior over all nodes, regardless of whether they are observable or latent. Therefore experiments will also be conducted with nodes that represent aggregate information (such as the aggregate spatial distribution of agent locations), thus making the agent locations themselves latent. This is much closer to a realistic application and, through the Bayesian network, an elegant way of incorporating different types of data.

5 Proposed Experiments and Expected Results

At the time of writing the experiments are under way. By the time of the workshop the paper will report initial experiments that explore: the ability of the framework to realistically model the true system state with full (albeit noisy) information about every agent; the impacts of reducing the number of observed agents (i.e. only observing a portion of the state vector); observing aggregate rather than individual-level data (i.e. all agent locations are latent).

[4] Keanu: https://github.com/improbable-research/keanu

References

1. Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature **525**(7567), 47–55 (2015). https://doi.org/10.1038/nature14956
2. Ghahramani, Z.: Bayesian non-parametrics and the probabilistic approach to modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **371**(1984), 20110553–20110553 (Dec 2012). https://doi.org/10.1098/rsta.2011.0553
3. Ghahramani, Z.: Probabilistic machine learning and artificial intelligence. Nature **521**(7553), 452–459 (May 2015). https://doi.org/10.1038/nature14541
4. Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press (2003)
5. Malleson, N., Ward, J.A., Heppenstall, A., Adcock, M., Tang, D., Coello, J., Crols, T.: Understanding Input Data Requirements and Quantifying Uncertainty for Successfully Modelling 'Smart' Cities. In: 3rd International Workshop on Agent-Based Modelling of Urban Systems. Stockholm, Sweden (2018)
6. Wang, M., Hu, X.: Data assimilation in agent based simulation of smart environments using particle filters. Simulation Modelling Practice and Theory **56**, 36–54 (2015). https://doi.org/10.1016/j.simpat.2015.05.001
7. Ward, J.A., Evans, A.J., Malleson, N.S.: Dynamic calibration of agent-based models using data assimilation. Royal Society Open Science **3**(4) (2016). https://doi.org/10.1098/rsos.150703