Real-time Pedestrian Modelling: Implementing the Ensemble Kalman Filter for an Agent-Based Model

Keiran Suchak — 200888140

May 28, 2019

Contents

List of Figures

List of Tables

Notes

- What is data assimilation?
- What is the Ensemble Kalman Filter?
- What are Agent-Based Models?
- How do people model pedestrians?
- EnKF: options for generating observations:
  - External (the old way):
    - observations come from a previous run of the model
    - therefore, they are independent
    - the problem here is that the ensemble members probably have been set with the wrong entrances and exits for agents
    - this is likely more realistic, ie when modelling people we ultimately won't know where they intend to go
    - but I don't think that we can overcome this without parameter estimation DA.
  - Internal (the new way):
    - observations come from the `base_model`
    - all ensemble members are deep copies of the `base_model`
    - consequently, they have the same parameters and initial conditions
    - the problem with this is that is not realistic
    - but we're going to do it anyway (for now)
- What happens when an agent in one of the ensemble member models tries to leave the station early? Because we're only doing state estimation (meaning just the agent coordinates), the agent won't get reactivated.
- Errors should converge on something relating to twice the exit distance.

Structure

- What is the problem?
- Why should people care?
- What has been done in the past to try to solve the problem?
- What are we doing to solve the problem? How is this different?
- How well does our attempt work? How does this compare to others?

Chapter 1

Introduction

A better understanding of how people move around their environment is of great utility to both academics and policy-makers. Such knowledge can be made use of in the contexts of urban planning, event management and emergency response, particularly when considering urban environments. Furthermore, this may also be of use to those interested in the social issues of mobility, inclusivity and accessibility of opportunities.

When considering such concepts, investigators often make use of modelling techniques. At their most fundamental, models represent our understanding of the system that we are studying — an understanding that may not be perfect (Stanislaw 1986). There exist modelling techniques for the simulation of how pedestrians move around urban spaces. However, these methods exist largely in isolation of the real-world — that is to say that whilst the simulations aim to reflect the real-world, there is no method by which we can incorporate up-to-date observations into these models to stop their divergence from reality.

Simulating pedestrian behaviour is often undertaken at the micro-scale, with such models typically aiming to model at the individual level or on a spatially fine-grained grid (Burstedde et al. 2001). One of the most prevalent simulation methods in this field is that of Agent-Based Modelling. Such methods consist of two key components: agents and environments. In an Agent-Based Model, we prescribe sets of rules by which individuals interact with each other and their local environments; as interactions take place on the micro-scale, we typically observe the emergence of structure at the macro-scale such as crowding (Batty et al. 2003) or lane formation (Liu et al. 2014). The evaluation of these rules is often not deterministic and instead introduces some element of randomness; these stochastic elements aim to emulate the variability of human behaviour. The introduction of such randomness in conjunction with an imperfect understanding of the phenomena

at play, however, typically result in simulation runs diverging from the real system.

In constructing their models, agent-based modellers undertake a development process that involves model verification, validation and calibration. We can take these to mean the following:

- **Model verification**: the process of ensuring that the implementation is an accurate representation of the model (Xiang et al. 2005).
- **Model validation**: the process of ensuring that the chosen model is an accurate representation of the phenomenon that we wish to study (Crooks et al. 2008).
- **Model calibration**: the process of searching for model parameter values such that we can achieve validation (Thiele et al. 2014).

Beyond this, modellers also make efforts to ensure that the initial model conditions are realistic by setting them based on historical data.

The practices of validation, calibration and setting initial model states based on historical data are appropriate for offline evaluations such as testing designs of new buildings or experimenting with different individual behaviours; however, when aiming to simulate events in real-time, this simply delays the inevitable divergence of the model from the real system. Furthermore, model parameters may be transient and thus require to be updated as time passes and the dynamics evolve.

Given the apparently inevitable divergence of stochastic simulations from the real systems that they aim to model, one may alternatively turn to big data. Data is now being generated in higher volumes and at greater velocity than ever before (Chen et al. 2014); however, there also exist issues with observation data from such systems. Whilst models typically allow us to simulate a whole system, observations are typically sparse in either time or space (or both); this is to say that observations rarely provide complete coverage of the events. We therefore seek a solution whereby we can integrate up-to-date observations into our models as the models continue to simulate the system.

One of the methods by which we can combine knowledge represented by our model with observations as they become available is through data assimilation techniques, which are most commonly used in the field of numerical weather prediction (Kalnay 2003). Such techniques are typically made up of two steps:

1. **Predict**: Run the model forward, estimating the state of the system, until new observations become available.

2. **Update**: Upon receipt of new observations, combine the model's estimate of the system state with the new data.

These steps are repeated iteratively in a cycle. It is important to note that just as the there is error associated with the model, we also acknowledge that there is observational error associated with the data. The aim of incorporating the observations into the model is to improve the model accuracy with respect to the true system state.

A large volume of work exists in which such techniques are applied to meteorological systems where the models used are based on differential equations. Significantly less work exists in which data assimilation methods are applied to agent-based models — in particular pedestrian models. This dissertation therefore aims to expand on the pre-existing work by implementing a data assimilation scheme known as the Ensemble Kalman Filter in conjunction with a relatively simple agent-based model of pedestrians crossing a two-dimensional station from one side to the other.

The remainder of this dissertation will provide an overview of some of the work that has been undertaken thus far on implementing data assimilation schemes with agent based models (Chapter 2), elaborate on how data assimilation works — particularly focussing on the aforementioned Ensemble Kalman Filter (Chapter 3) — and examine the effectiveness of this method in improving the accuracy of the agent-based model (Chapter 4).

Chapter 2

Literature Review

This is the literature review.

There has been some work on this stuff (Wang & Hu 2015, Ward et al. 2016).

2.1 Data Assimilation with Agent-Based Models

- (Ward et al. 2016)
- (Wang & Hu 2015)
- (Rai & Hu 2013)
- ()

2.2 Data Assimilation with Cellular Automata

-

Chapter 3

Method

- Intro to data assimilation
- What is data assimilation?
- Where does it come from?
- What is the point?
- What types of data assimilation are available to us?
- Which one are we going to use?

The updating of the model state is undertaken on the basis of Bayes Rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.1}$$

Bayes Rule is made up of four components:

1. $P(A)$: Prior
2. $P(A|B)$: Posterior
3. $P(B|A)$: Likelihood
4. $P(B)$: Marginal likelihood

Then let's frame our problem notationally — this is what Bayes theorem looks like for our situation:

$$P(\mathbf{x}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{x})\,P(\mathbf{x})}{P(\mathbf{d})} \tag{3.2}$$

In this case, each of the components are taken to mean

1. Prior: The probability of the model state
2. Posterior: The probability of the model state given the data
3. Likelihood: The probability of the data given the model state
4. Marginal likelihood: The probability of the data

1. $P(A)$: Prior
2. $P(A|B)$: Posterior
3. $P(B|A)$: Likelihood
4. $P(B)$: Marginal likelihood

3.1 Kalman Filter

There's a lovely data assimilation method called the Kalman Filter (Kalman 1960).

- What is the Kalman Filter?
- How does it work?
- When is it good?
- When is it bad?
- What can we do to improve it?

3.2 Ensemble Kalman Filter

Problems with the Kalman Filter:

- it assumes Gaussian PDFs
- it assumes linear model
- Cost of evolving covariance matrix

In order address some of these problems, the Ensemble Kalman Filter was developed (Evensen 2003, 2009), which acts as an approximation of the Kalman Filter. This approximation is achieved by using an ensemble of sample state vectors to represent the state distribution. As such, the state is represented as follows:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] = [\mathbf{x}_i], \quad \forall i \in (1, N), \tag{3.3}$$

where the state ensemble matrix, $\mathbf{X}$, consists of N state vectors, $\mathbf{x}_i$. Similarly, the observations are represented as follows:

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] = [\mathbf{d}_i], \quad \forall i \in (1, N), \tag{3.4}$$

with each member of the data ensemble matrix, $\mathbf{D}$, being the sum of the original observation $\mathbf{d}$, and a random vector, ϵ_i:

$$\mathbf{d}_i = \mathbf{d} + \epsilon, \quad \forall i \in (1, N). \tag{3.5}$$

The random vector is drawn from an unbiased normal distribution:

$$\epsilon \sim \mathcal{N}(0, \mathbf{R}). \tag{3.6}$$

$$\hat{\mathbf{X}} = \mathbf{X} + \mathbf{K}(\mathbf{D} - \mathbf{H}\mathbf{X}) \tag{3.7}$$

$$\mathbf{K} = \mathbf{Q}\mathbf{H}^T\left(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{3.8}$$

3.2.1 Different Types of Ensemble Kalman Filter

Talk about the different types of EnKF and the implications for ensemble size (Keller et al. 2018).

Chapter 4

Results

These are the results.

Chapter 5

Conclusion

This is the conclusion.

Appendix A

Code Documentation

This is where I explain the design choices for how the enkf was coded.

Bibliography

Batty, M., DeSyllas, J. & Duxbury, E. (2003), 'The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades', *International Journal of Geographical Information Science* **17**(7), 673–697.

Burstedde, C., Klauck, K., Schadschneider, A. & Zittartz, J. (2001), 'Simulation of pedestrian dynamics using a two-dimensional cellular automaton', *Physica A: Statistical Mechanics and its Applications* **295**(3-4), 507–525.

Chen, M., Mao, S. & Liu, Y. (2014), 'Big data: A survey', *Mobile networks and applications* **19**(2), 171–209.

Crooks, A., Castle, C. & Batty, M. (2008), 'Key challenges in agent-based modelling for geo-spatial simulation', *Computers, Environment and Urban Systems* **32**(6), 417–430.

Evensen, G. (2003), 'The ensemble kalman filter: Theoretical formulation and practical implementation', *Ocean dynamics* **53**(4), 343–367.

Evensen, G. (2009), 'The ensemble kalman filter for combined state and parameter estimation', *IEEE Control Systems Magazine* **29**(3), 83–104.

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Journal of basic Engineering* **82**(1), 35–45.

Kalnay, E. (2003), *Atmospheric modeling, data assimilation and predictability*, Cambridge university press.

Keller, J., Hendricks Franssen, H.-J. & Marquart, G. (2018), 'Comparing seven variants of the ensemble kalman filter: How many synthetic experiments are needed?', *Water Resources Research* **54**(9), 6299–6318.

Liu, S., Lo, S., Ma, J. & Wang, W. (2014), 'An agent-based microscopic pedestrian flow simulation model for pedestrian traffic problems', *IEEE Transactions on Intelligent Transportation Systems* **15**(3), 992–1001.

Rai, S. & Hu, X. (2013), Behavior pattern detection for data assimilation in agent-based simulation of smart environments, *in* '2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)', Vol. 2, IEEE, pp. 171–178.

Stanislaw, H. (1986), 'Tests of computer simulation validity: what do they measure?', *Simulation & Games* **17**(2), 173–191.

Thiele, J. C., Kurth, W. & Grimm, V. (2014), 'Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using netlogo and r', *Journal of Artificial Societies and Social Simulation* **17**(3), 11.

Wang, M. & Hu, X. (2015), 'Data assimilation in agent based simulation of smart environments using particle filters', *Simulation Modelling Practice and Theory* **56**, 36–54.

Ward, J. A., Evans, A. J. & Malleson, N. S. (2016), 'Dynamic calibration of agent-based models using data assimilation', *Royal Society open science* **3**(4), 150703.

Xiang, X., Kennedy, R., Madey, G. & Cabaniss, S. (2005), Verification and validation of agent-based scientific simulation models, *in* 'Agent-directed simulation conference', Vol. 47, p. 55.