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

Keiran Suchak — 200888140

May 29, 2019

Contents

List of Figures

List of Tables

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

2.1 Model Calibration

As touched upon in Chapter 1, the process of developing an agent-based model typically involves some form of model calibration. Model calibration is the procedure of fine-tuning the model that we are using such that it best fits the particular situation that we are seeking to model (Crooks & Heppenstall 2012). There are a large number of different manners in which we can calibrate agent-based models (Thiele et al. 2014).

- MORE ABOUT MODEL CALIBRATION

2.2 Data Assimilation

- DATA ASSIMILATION IN GENERAL
- HOW THIS RELATES TO CALIBRATION

The process of data assimilation involves making use of observations along with prior knowledge (which, in our case, is encoded in a model) to produce increasingly accurate estimates of variables of interest. Such a process can be achieved through a Bayesian filtering approach (Grewal et al. 1995).

2.2.1 Data Assimilation with Agent-Based Models

- Ward et al. (2016) — model of pedestrians on Briggate, a 1-D strip along which pedestrians walk, using enkf for both state and parameter calibration.

- Wang & Hu (2013, 2015) — agents occupying a smart environment/building with a view to modelling population density, using particle filter.
- Rai & Hu (2013) — agents occupying a smart office/building, using particle filter, extends Wang & Hu (2013, 2015) by incorporating a Hidden Markov Model for behaviour pattern detection.
- Wang et al. (2017) — model of maritime pirates, using random finite set based data assimilation instead of kf or pf — why? does this have any relevance?

2.2.2 Data Assimilation with Cellular Automata

Whilst this dissertation focuses on the application of data assimilation methods to agent-based models, there also exists a body of work that makes use of the same methods in conjunction with cellular automata.

- WHAT ARE CELLULAR AUTOMATA
- HOW DO THEY RELATE TO ABMS?
- Li et al. (2017) — CA for urban land use, using enkf.
- Li et al. (2012) — CA for urban land use, using enkf.
-

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

3.1 Kalman Filter

The Kalman Filter (Kalman 1960) is the best linear unbiased estimator, minimising posterior mean squared error.

- What is the Kalman Filter?
- How does it work? Sequential data assimilation to update state and covariance matrix, weighted based on model forecasts and observations.
- When is it good? Optimal solution for linear model and Gaussian errors.
- 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:

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

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

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

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).

- Damping: counteract filter divergence
- Localisation: reduce the effect of spurious correlations
- Hybrid EnKF: Covariance matrix is made up of the weighted sum of the usual covariance matrix and a separate static covariance matrix that encodes prior underlying knowledge about the system
- Dual EnKF: Split the state vector into state and parameters. At assimilation: update parameters, recalculate forecast, update state
- Normal Score EnKF: Developed to handle non-Gaussian PDFs in EnKF. At assimilation: transform state, parameters and measurements into Z-scores, perform EnKF update based on transformed values, transform back from Z-scores
- Iterative EnKF

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.

- options for generating observations:
  - external
    - observations come from a previous model run as synthetic data
    - therefore they should be independent
    - the problem here is that the ensemble members probably have been set with the wrong entrances and exits or agents
    - this is likely more realistic, because when modelling people we ultimately won't know where they intend to go
    - but we may be able to overcome this using parameter estimation DA.
  - internal
    - 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
    - problem here is that it's not very realistic
    - but we're going with it for this particular version

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.

Crooks, A. T. & Heppenstall, A. J. (2012), Introduction to agent-based modelling, *in* 'Agent-based models of geographical systems', Springer, pp. 85–105.

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.

Grewal, M. S., Andrews, A. P. & Bass, R. (1995), 'Kalman filtering: Theory and practice', *IEEE Transactions on Automatic Control* **40**(11), 1983.

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.

Li, X., Lu, H., Zhou, Y., Hu, T., Liang, L., Liu, X., Hu, G. & Yu, L. (2017), 'Exploring the performance of spatio-temporal assimilation in an urban cellular automata model', *International Journal of Geographical Information Science* **31**(11), 2195–2215.

Li, X., Zhang, Y., Liu, X. & Chen, Y. (2012), 'Assimilating process context information of cellular automata into change detection for monitoring land use changes', *International Journal of Geographical Information Science* **26**(9), 1667–1687.

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. (2013), Data assimilation in agent based simulation of smart environment, *in* 'Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation', ACM, pp. 379–384.

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.

Wang, P., Li, G., Ju, R. & Peng, Y. (2017), 'Random finite set based data assimilation for dynamic data driven simulation of maritime pirate activity', *Mathematical Problems in Engineering* **2017**.

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.