Forecasting Short-Term Urban Dynamics: Data Assimilation for Agent-Based Modelling

Nick Malleson, Alice Tapper, Jon Ward, Andrew Evans

March 29, 2017

Abstract

Background. The 'data deluge', coupled with related 'smart cities' initiatives, have led to a proliferation of models that capture the current state of urban systems to a high degree of accuracy. However, their ability to *forecast* future system states appears limited. Agent-based modelling (ABM) is well suited to modelling urban systems, but at present the methodology is seriously limited in its ability to incorporate up-to-date data (such as social media contributions, mobile telephone activity, public transport use, etc.) as they arise in order to reduce uncertainty in model forecasts.

Methods. This paper presents ongoing work that adapts data assimilation techniques from fields such as meteorology in order to allow agent-based models to be optimised using streaming data *in real time*. Here, a simple example of an agent-based model used to simulate the movement of people as they travel along a street is illustrated. Importantly, the model is optimised dynamically with an ensemble Kalman filter in response to hypothetical pedestrian count data.

Findings. The data assimilation technique reliably estimates the model parameter that it is attempting to optimise. Surprisingly, however, the estimates of the 'true' system state that are produced by the model combined with noisy observations are less accurate than the observations in isolation. This is likely an artefact of the specific system under study here. Ultimately, we work towards a combination of ABM and data assimilation methods that will be able to assimilate streaming 'smart cities' data into models in real time.

Keywords: agent-based modelling; data assimilation; data-driven simulation; urban mobility; ensemble Kalman filter.

1 Introduction

The Big Data "revolution" (Mayer-Schönberger and Cukier, 2013) has lead to the emergence of abundant, high-resolution, individual-level data about peoples' daily activities. This "data deluge" (Kitchin, 2013a) has subsequently spawned interest in 'smart cities'; a term that is commonly used to refer to cities that "are increasingly composed of and monitored by pervasive and ubiquitous computing" (Kitchin, 2013b). However, one aspect to many smart cities, that is largely absent in the published literature, is the ability to forecast as well as react. Whilst most initiatives inject real-time data, these data are rarely used to make real-time predictions about the future (see, for example, Bond and Kanaan, 2015; Yamamoto, 2015; Gray et al., 2016). This might be due to the proprietary nature of many initiatives, which are often designed and implemented by corporations rather public bodies, but it is equally likely that a lack of appropriate methods is at fault. Although 'black box' artificial intelligence methods are progressing rapidly there is little evidence that these are being used to forecast future states of smart cities.

Agent-based modelling (ABM) is well suited to modelling urban systems and therefore might be an ideal methodology to fill this predictive gap. However, ABMs are seriously limited in their ability to incorporate up-to-date data as they arise in order to reduce uncertainty in model forecasts (Lloyd et al., 2016; Ward et al., 2016). Models are typically calibrated once, using historical data, and then projected forward in time to make a prediction. As the systems under study are usually complex, models diverge rapidly from reality; weakening the accuracy of their forecasts.

This paper presents ongoing work that adapts data assimilation techniques from fields such as meteorology (Bauer et al., 2015) in order to allow agent-based models to be optimised using streaming data *in real time*. The main difficulty that the research faces is that the methods are typically designed for systems of differential equations and cannot easily be disassociated from their models. Here, we present a simple example of an agent-based model used to simulate the movement of people as they move down a street, optimised dynamically through hypothetical pedestrian count data that could emerge from a footfall camera counter (these are typically used by policy makers to quantify the movements of people around urban areas).

2 Context

Ultimately, the aim of this research is to develop methods (both agent-based models and data assimilation techniques) that can be used to develop a city-wide model of urban flows using data that are being streamed from the abundant sensor networks that characterise smart cities to suppress error in real time. At this early stage, however, we focus on a much more tightly constrained scenario. In the city of Leeds, UK, policy makers are interested in encouraging visitors who attend the city centre to explore an area to the south of the centre that is only a short walking distance (approximately 10 minutes) but under utilised. To this end, cameras have been installed at the entrance and exit of a largely pedestrianised thoroughfare to count the number of passers-by and better understand how the link is being used. However, it is possible for people to leave and enter the street at various midpoints (in between the camera locations) without being counted. Therefore we work towards an agent-based model that simulates the movements of pedestrians and is capable of estimating the total volume of people present in the street at any one time, using streaming camera from the data as a means of dynamically suppressing error, here associated with estimating the losses to other routes.

3 Methods

3.1 The Agent-Based Model

To reflect the simple scenario outlined in Section 2, a simplified agent-based model is produced. Figure 1 outlines the environment. It consists of an entrance point (point A, coordinates 0,1) and an exit point (point B, coordinates 0,40), separated by one-dimensional 'street'. *Active* agents move from point A to point B, moving one coordinate at each iteration of the model. It is a requirement of the data assimilation technique used (discussed in Section 3.2) that the underlying model is Markovian, such that future model states can be predicted entirely from the current model state. In this manner a model instance can be expressed in its entirety as a single state vector. The implication here is that all agents who could potentially enter the system through point A must exist in a dormant state for the times that they are not actually in the system. Therefore the total number of agents in the model is constant, but at any given time a number of agents will have been *retired* (i.e. they are present in the model, but not active in the simulated street). When an agent is not active, it will be located at a null point N. Fortunately in this simple model we know the total number of agents who could possibly be active at any time (600, as discussed in the following section) and so can guarantee not to 'run out' of dormant agents. In a system that did not have this *a priori* information, the Markovian restriction would be more problematic, although probably not insurmountable.

All active agents will be on one of the 40 cells that make up the street. Hypothetical footfall cameras are positioned at points A and B to record the number of agents entering (in the case of A) or leaving (in the case of B) at these positions respectively. The cameras return hourly footfall counts. Therefore the model

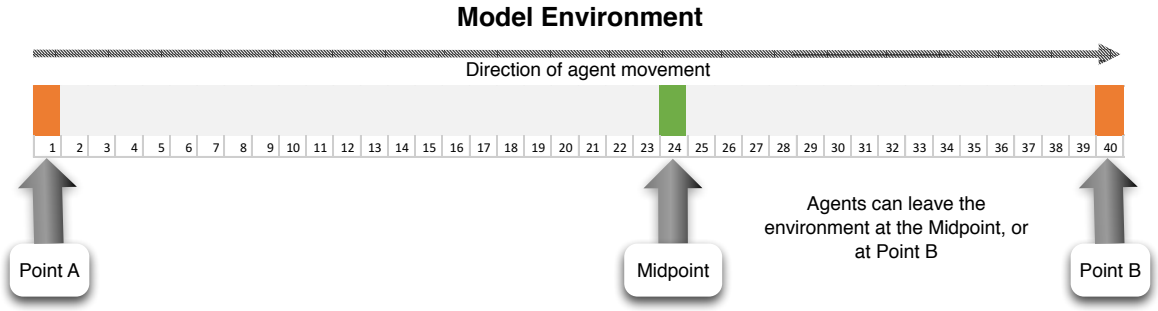

Figure 1: The agent-based model environment.

state vector contains: the location of each agent, the bleed out rate (the only model parameter, discussed below), and the current camera counts at point A and B. With a system containing 600 agents, each of which has two parameters to represent their location (x and y)[1] this is a 1203×1 column vector.

Each iteration of the simulation corresponds to one minute of real time. Every 60 iterations (minutes) a certain number of agents are activated and moved from point N to point A. In every subsequent iteration these agents then move to the next coordinate in the street. To simulate the ability of agents to leave the street without being captured by the camera at point B, a mid-point M exists approximately in the middle of the street (coordinates 0, 24). On reaching M, some of the agents will instantly return to the null point N (i.e. they leave the environment) with a certain probability termed the *bleed out rate*, r. Those that do not leave at M will continue their journey to point B. Once they reach point B, they have effectively left the system, so return to the null point N.

The number of agents being activated at point A is set for each hour of the day, and does not vary day to day. The distribution throughout the day has been approximated as a normal distribution, peaking at midday. Figure 2 shows an example run of 5 days. The blue line represents camera A counts at point A, and the green line represents are camera B counts at point B. In this simple case the bleed out rate, $r \approx 0.5$, so roughly half the agents leave through the midpoint rather through through point B. Hence the agent count at camera B is approximately half of that of A over an hour.

To test the methods, we will assume (unrealistic) knowledge of the distribution of agents being activated at point A, and attempt to use data assimilation to estimate the true number of agents at point B – thereby performing parameter estimation of the bleed out rate. To perform the data assimilation, an ensemble Kalman filter is used, as discussed in the following section.

[1]Note that in this one-dimensional application, the agents' y coordinate will always be 0, but we include the parameter in order to ease the transition into a more complex, two-dimensional model

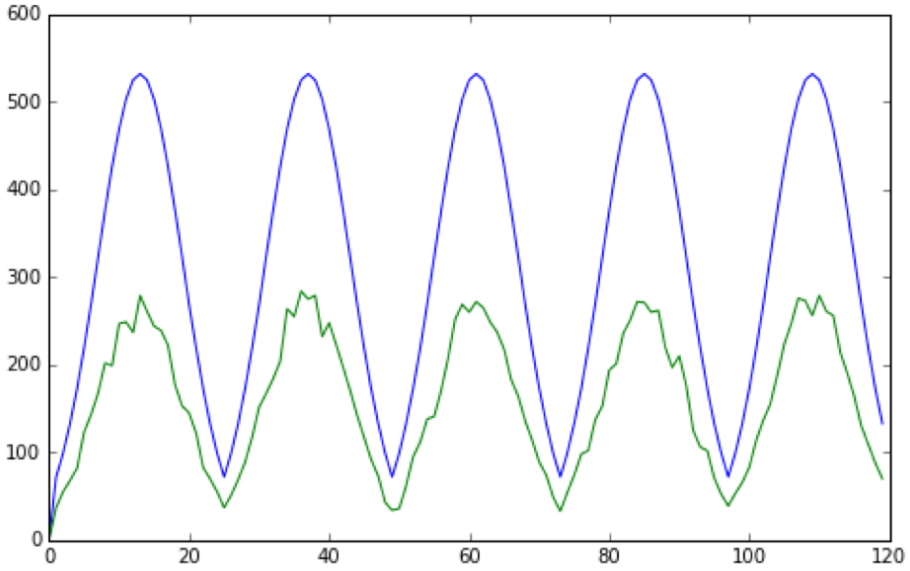

Figure 2: An example model run of 5 simulated days. The blue counts are camera A counts at point A, and the green counts are camera B counts at point B.

3.2 The Ensemble Kalman Filter (EnKF)

The ensemble Kalman filter (EnKF) is a standard method used in data assimilation in fields such as meteorology (Lewis et al., 2006; Kalnay, 2003). The algorithm combines a series of noisy observations made over time with a model representing the underlying system which is initialised with prior real-world data to produce estimates of the *true* state of the system. The noisy observations are combined with model forecasts, and their influences on the adjusted estimates are weighted relative to the uncertainty of each. The adjusted estimates produced should ideally be more accurate than the noisy observations themselves. It should be noted that this paper does not attempt to reproduce the underlying mathematics in full. For this, the interested reader can refer to any number of alternative sources such as Kalnay (2003), Lewis et al. (2006), or Ward et al. (2016).

The EnKF maintains a set (*ensemble*) of estimated model states. This ensemble of state vectors is initialised either with knowledge, or estimates, of the original system state and its uncertainty. Each model state is then evolved forward independently according to two steps:

The forecast step:

The EnKF runs each model forward until the next observation time – i.e. the time at which observational data (camera counts) become available. Here these counts become available once every hour. These model runs generate a set of forecasts of the current system state (each model estimates the current state of the system). The mean of the ensemble forecast states will give an estimated

forecast of the true state, usable for any immediate forecasting needs, while the covariance of the ensemble forecast states provides a measure of its uncertainty.

The data assimilation step:

Upon receiving the actual observation (camera counts), the ensemble forecasts are updated accordingly. The updated values are called the ensemble analysis. The ensemble analysis mean will give the best estimation of the true state, while the covariance of the ensemble analysis states provides a measure of its uncertainty.

These steps are repeated each time a new observation is made. Because model parameters are included in the state vector, these parameters are also updated during the iterative process through their covariance with the observables. Predicting these unknown parameters this way is known as sequential parameter estimation.

3.3 Application of the EnKF to the ABM

Hypothetical 'Real World' Data

As this ongoing work (see Ward et al., 2016) represents the first test of an agent-based model coupled with an ensemble Kalman filter in order to assimilate real-time data, the methodology was kept simple. A synthetic set of 'streaming' camera count data was generated. The bleed out rate (r) was drawn once from a normal distribution with $\mu = 0.5$ and $\sigma^2 = 0.1$ and subsequently kept constant. The model was executed for 7,200 simulated minutes (5 days). This 'truth' model was used to create hypothetical 'real world' camera counts which will be used throughout the experiments to assess model errors and, ultimately, the success of the data assimilation process. In reality, these observations would have an element of uncertainty associated with them; no measuring instrument is going to be perfect. Therefore normal random noise was added to each of the true camera counts to simulate the error that would occur in real-world observations. This noise was drawn from a normal distribution with $\mu = 0$ and $\sigma^2 = 5$ (although adjusted to $\sigma^2 = 10$ and $\sigma^2 = 15$ in later runs of the Kalman filter to explore the impacts of different levels of noise).

The Forecast Step

To conduct data assimilation, new real-world observations are incorporated with model estimates of the system. Therefore the 'true' camera counts used here (as produced by our 'true' simulation) can be represented as an *observation vector* (in this case a 2×1 column vector), which consists of the number of agents who passed cameras A and B in the previous hour.

The ensemble of independent model instances was initialised by drawing 30 bleed out rates (r) from the same normal distribution that the 'true' parameter was chosen from ($\mu = 0.5$ and $\sigma^2 = 0.1$). The models were then run forward a simulated hour for each of these rates. The resulting 30 state vectors (the *forecasts*) were stored. The ensemble mean was calculated, and the variance of

each value informed a covariance matrix P. At this point we have the forecast mean (including the forecast camera B counts), and the forecast variance.

The Data Assimilation Step

Now we assume that 1 hour of real-world time has passed and the footfall cameras have produced new footfall data covering the previous hour. Therefore the data assimilation step begins. An observation vector (the 'real' counts from cameras A and B) is extracted from the truth data.

The matrix H is a transformation matrix that changes the state vector into the same form as the observation vector. In this example, with a 2×1 observation vector and a 1203×1 state vector, it is a 2×1203 matrix with 1s in the correct positions to pick the camera counts out of the state vector. The next step in the assimilation process is to calculate the Kalman gain matrix, K. This represents the importance of the observational estimates relative to the predicted estimates. If the observations are accurate (have low uncertainty) then they will be more important in estimating the final state than if they had high uncertainty. Following this, the *ensemble analysis* begins. Recall that the ensemble is the set of 30 model state vectors; these are adjusted to take account of the new observations from the 'real world'. The ensemble analysis is calculated as follows:

$$Analysis = Forecasts + K \times (VirtualObservations - Forecasts)$$

We can now calculate the analysis mean, and using this work out the analysis variance. As before, this informs the analysis covariance matrix.

... and repeat ...

Following the assimilation step, the independent models in the ensemble have been updated to incorporate estimates of the system from observational data. Therefore the new estimate of the true system state is a combination of model forecasts *and* observational data. Together, these should be a more accurate estimate of the real state than observations or model estimates in isolation. The models can now be forecasted forward an hour, and the resulting state vectors used provide the new forecasts. When new observational data become available the process repeats.

4 Results

This section attempts to identify how successful the data assimilation method was at both predicting the footfall at point B and correctly estimating the bleed out rate (r). In effect, does the combination of model forecasts and observational data get closer to the truth than the virtual observations do in isolation?

Figure 3 illustrates the results obtained when using a virtual observation error of $\mu = 0$, $\sigma^2 = 5$. In 3(a) only the forecast and the analysis are clearly distinguishable – the analysis appears to be so close to both the truth and the virtual observations that this level of resolution does not show any clear difference. 3(c)

shows the bleed out rate changing with time. The fact that the Kalman filter quickly settles on an accurate value for the bleed out rate is clearly reflected in the overall forecasts, which in 3(a) can be seen to mirror the ground truth well; the deviations from the normal distribution that occur due to stochasticity are naturally not picked up on by the ensemble, which will always ultimately average to an approximate normal distribution. 3(b) shows a zoomed-in view of a period when the parameter estimation is close to the truth. For a period of time here the analysis is closer to the ground truth than the virtual observations.

As hypothetical 'true' data were generated, it is possible to calculate the error associated with the forecast, analysis, and observational data independently. This is accomplished using the Root Mean Square Error (RMSE), where larger values are indicative of greater error. We find forecast RMSE: 9.80, analysis RMSE: 2.58 and observation RMSE: 0.91. This is a somewhat surprising result as we would expect the analysis RMSE to be less than the observation RMSE, as we would expect the model system to converge on the underlying real system even with a stochastic component being added to the observations; the filter dampening down the effect of the stochastic elements on the bleed out rate. One reason for this could simply be due to the level of the randomness of the model. Knowledge of the underlying system helps to guide knowledge of camera B, but ultimately the data rely so heavily on randomness that these intelligent guesses will rarely be closer than observations (even noisy observations).

This aside, the parameter estimation does well. After a period of initial fluctuation the bleed out rate settles to a roughly constant value, varying slowly from $0.427 - 0.435$. Given that the true bleed out rate is 0.433, this is fairly successful. Figure 4 shows the parameter estimation results for three runs, with virtual observation standard deviations 5, 10 and 15 respectively. All three do reasonably well, with no discernible trend in accuracy due to observation error.

5 Conclusions and Future Outlook

This paper has presented the most recent results in an ongoing programme of work (see Ward et al., 2016) towards adapting data assimilation methods for use in agent-based modelling. Specifically, we make use of an ensemble Kalman filter (EnKF), which is commonly used in meteorology and other fields, to perform parameter estimation on a simple agent-based model. Ultimately, we work towards a combination of ABM and data assimilation methods that will be able to assimilate streaming 'smart cities' data (e.g. social media contributions, pedestrian counters, mobile phone activity, etc.) into models *in real time*. This should help to substantially reduce the error in ABMs of urban systems and allow them to make more reliable forecasts. Data assimilation has played a key role in transforming the accuracy of numerical weather prediction (Kalnay, 2003), to the extent that 7-day weather forecasts are now more accurate than 5-day forecasts were in the 1990s (Bauer et al., 2015). We envisage a similar transformation for agent-based modelling if the complex assumptions associated with data assimilation methods can be overcome, and methods can be adapted to work with the more complex models that are typically representative of ABM as a whole.

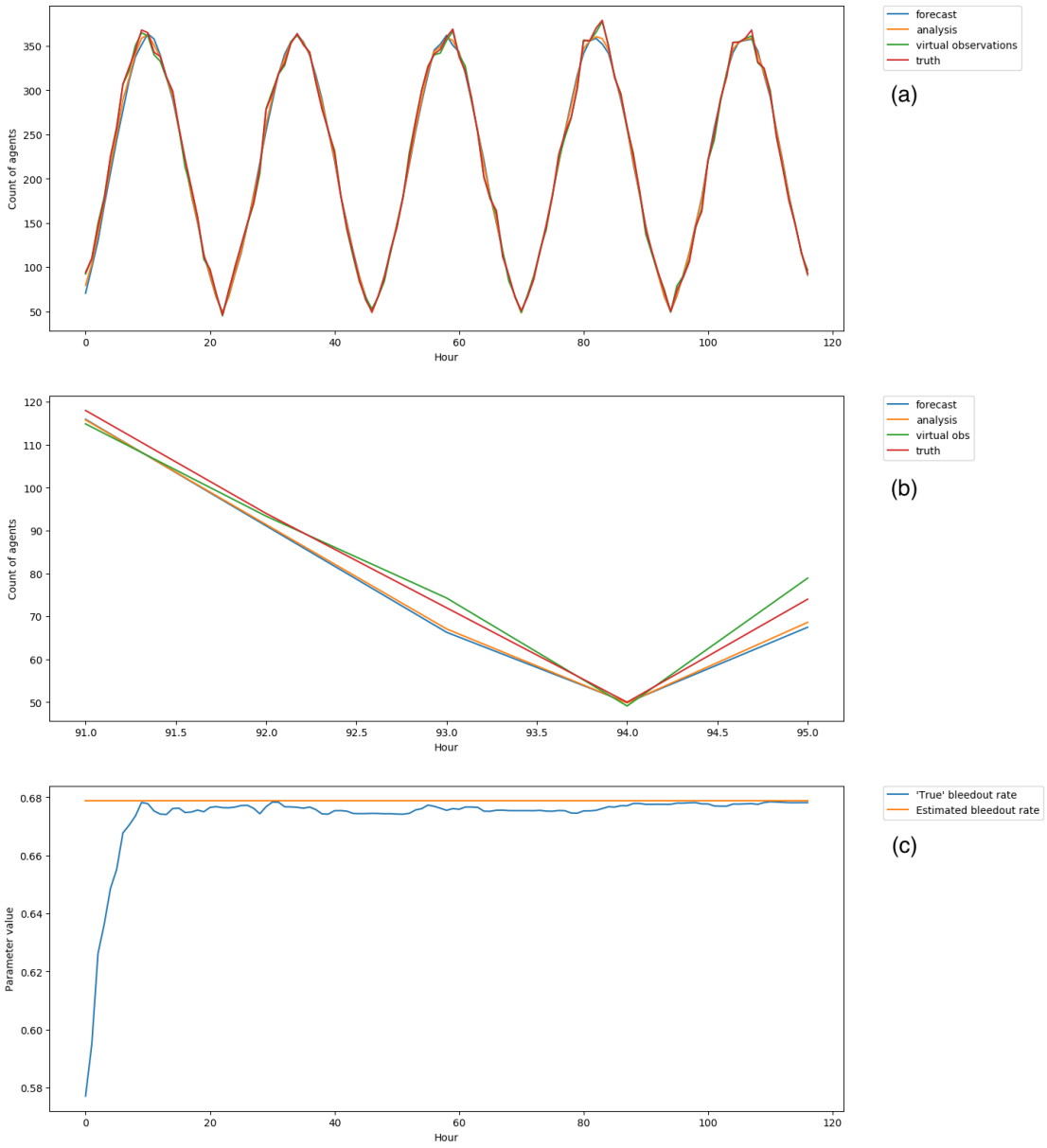

Figure 3: Graphs showing the Kalman filter results over 5 simulated days. (a) shows the number of agents who pass point B from the forecast, analysis, virtual observations and ground truth together. (b) focuses on one part of (a) to make clear the distinction between the four different time series. (c) shows the results of the sequential parameter estimation; comparing the 'true' bleed out rate (r, constant) that was used to generate the ground truth and the value of the bleed out rate in the state vector.

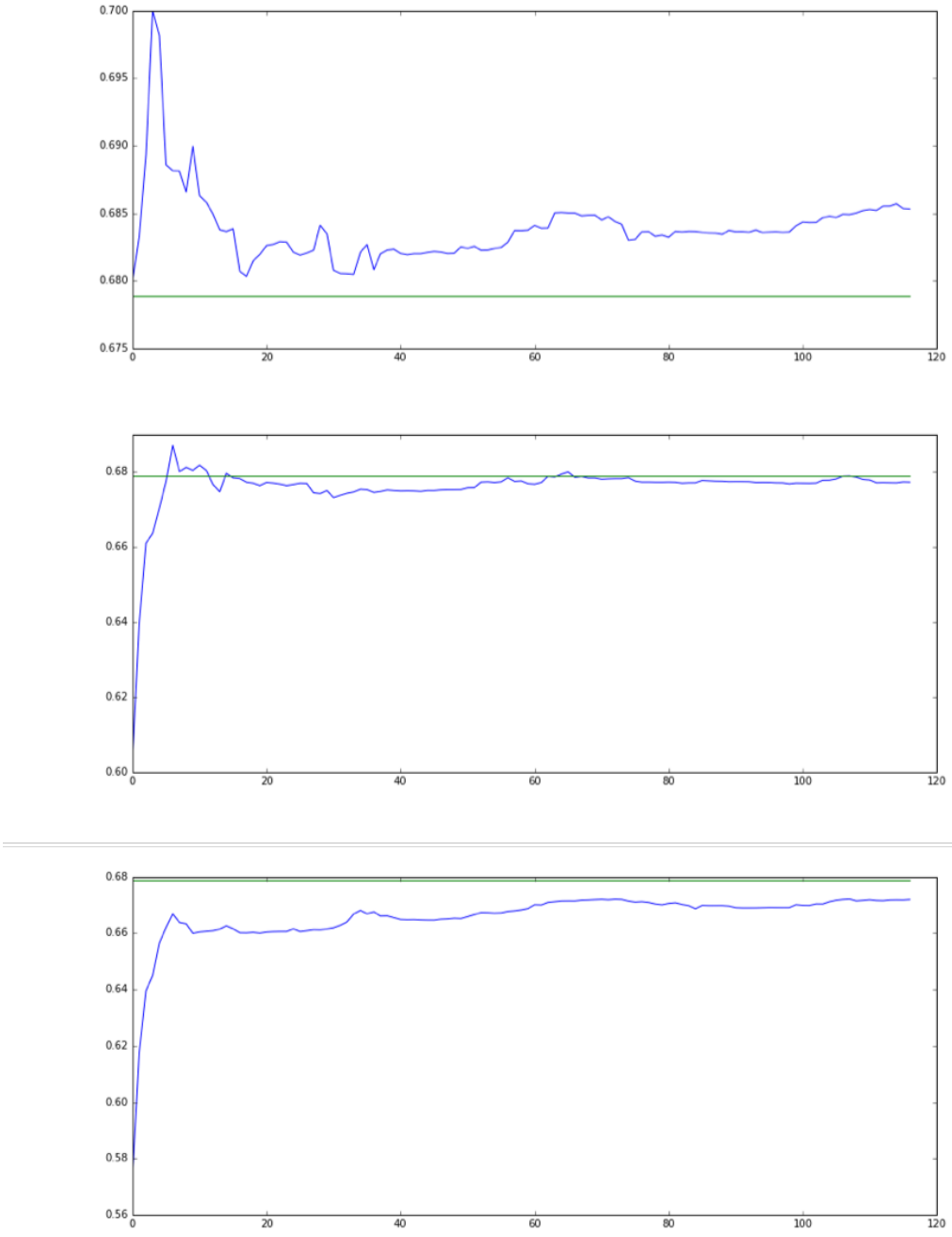

Figure 4: Graphs showing the sequential parameter estimation for virtual observations with error mean 0, standard deviation 5 (top), 10 (middle) and 15 (bottom). The 'true' parameter does not change). The parameter estimation appears to be the least accurate for the instance where the standard deviation is smallest.

References

Bauer, P., Thorpe, A., and Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55.

Bond, R. and Kanaan, A. (2015). MassDOT Real Time Traffic Management System. In Geertman, S., Ferreira, J., Goodspeed, R., and Stillwell, J., editors, *Planning Support Systems and Smart Cities*, pages 471–488. Springer International Publishing, Cham.

Gray, S., O'Brien, O., and Hügel, S. (2016). Collecting and Visualizing Real-Time Urban Data through City Dashboards. *Built Environment*, 42(3):498–509.

Kalnay, E. (2003). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press.

Kitchin, R. (2013a). Big data and human geography Opportunities, challenges and risks. *Dialogues in Human Geography*, 3(3):262–267.

Kitchin, R. (2013b). The Real-Time City? Big Data and Smart Urbanism. *SSRN Electronic Journal*.

Lewis, J. M., Lakshmivarahan, S., and Dhall, S. (2006). *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, Cambridge.

Lloyd, D. J. B., Santitissadeekorn, N., and Short, M. B. (2016). Exploring data assimilation and forecasting issues for an urban crime model. *European Journal of Applied Mathematics*, 27(Special Issue 03):451–478.

Mayer-Schönberger, V. and Cukier, K. (2013). *Big Data: A Revolution That Will Transform How We Live, Work and Think*. John Murray, London, UK.

Ward, J. A., Evans, A. J., and Malleson, N. S. (2016). Dynamic calibration of agent-based models using data assimilation. *Open Science*, 3(4).

Yamamoto, S. (2015). Development and Operation of Social Media GIS for Disaster Risk Management in Japan. In Geertman, S., Ferreira, J., Stillwell, J., and Goodspeed, R., editors, *Planning Support Systems and Smart Cities*, Lecture Notes in Geoinformation and Cartography. Springer International Publishing.