Parameter Estimation Using Data Assimilation For Time-Varying Parameters in Epidemiology

Ksenia Lepikhina

December 16, 2019

1 Background

The spread of influenza is complex. The infectious disease is known to mutate seasonally and within a single flu season making it challenging to study. Many methodologies[1][2][3] have been proposed to model the spread using different variations of the SIRS model. For example, Danon et. al. [1] use SIRS network models to track the spread of influenza while Cazelles as well as Yang use assimilation methodologies to determine the parameters of the SIRS model.

Epidemiology is a complex field which can be challenging to study with sparse datasets. In the absence of complete data, data assimilation methods provide a framework to reconstruct time dynamic processes which allow for an epidemic to be described. Simply by applying the methodologies to a stochastic model, it is possible to reconstruct complex epidemics over certain periods of time.

2 Summary

The goal of this paper is to learn how data assimilation methodologies can be applied to parameter estimation problems in particular for an SIRS dynamic model. This paper primarily studied the methods suggested in Cazelles et. al. [2] and chose to focus on the MCMC/data assimilation portion. Cazelles implements the Extended Kalman Filter in order to estimate the likelihood of the SIRS model parameters and this paper aims to estimate parameters for the SIRS model using the MCMC to converge to the most likely parameters using the likelihood. The likelihood in this paper will be found using the ETKF (Ensemble Transform Kalman Filter).

The purpose of this study is to learn about parameter estimation with Markov chain Monte Carlo and how data assimilation can be used to estimate a likelihood for the algorithm. This paper will implement a ”toy” model by simulating an SIRS model with known parameters and try to estimate the true parameters using the aforementioned MCMC/ETKF. It will then try to implement the same algorithm on a selection of the CDC influenza dataset.

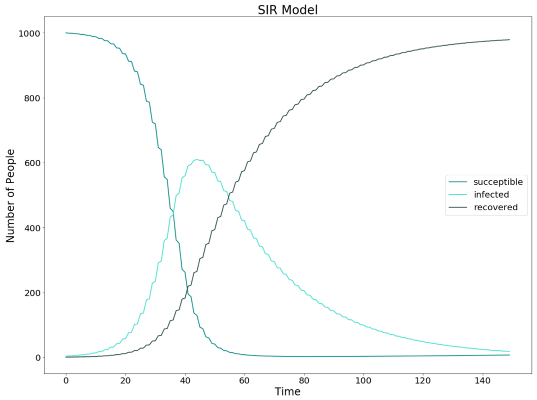
A successful implementation would allow for accurate predictions of peaks in the flu season as well as provide reasonable estimates for how many people would be affected by an epidemic. This would inform public health policy and individual behavior.

3 Introduction

To begin, this paper would like to give a brief introduction to the SIRS model it chose to study. There are multiple different SIRS models as there is no correct model. The SIRS model is a model that describes the movement over time from the susceptible class of people to infected, then recovered and back to susceptible. The reason for choosing SIRS instead of SIR or SI (etc.) for this paper is because is cyclical seasonally. This model theoretically assumes a constant population meaning that the total population is the sum of the three categories. The SIRS set up I chose to use follows the following dynamic system:

$$\begin{aligned}
\frac{dS}{dt} &= \mu(N-S) - \frac{\beta(t)SI}{N} + \alpha R\\
\frac{dI}{dt} &= \frac{\beta(t)SI}{N} - (\gamma+\mu)I\\
\frac{dR}{dt} &= \gamma I - (\alpha+\mu)R\\
\beta(t) &= \beta_0\left(1+\beta_1 \sin\left(\frac{2\pi t}{365} - 0.4\pi\right)\right)
\end{aligned}$$

In the system of differential equations above, S, I and R are state variables where S is the number of susceptible people, I is the number of infected and R is the number of recovered. N is the constant population. The parameters that this paper tries to estimate are μ, β_0, β_1, α and γ. Here, the $\beta(t)$ is a time varying parameter that represents *the transmission rate.* It is sinusoidal because it takes into account the fact that the flu may increase in severity over time and that people may be more susceptible to infection during the peak flu season due to weaker immune systems. β_0 and β_1 are parameters that need to be estimated for the transmission rate. μ represents the *mortality rate* which is a measure of the number of deaths in a particular population. α is the *average duration of immunity* and γ is the *recovery rate.*



The figure above is a simple visualization of the dynamic system where the dark turquoise line is the number of susceptible people, the cyan line is the number of infected people and the black line is the number of recovered people.

4 Methods

As mentioned in earlier sections this paper discusses a method to estimate parameters using a combination of the MCMC and ETKF data assimilation. The following paragraphs will first detail the broad MCMC for parameter estimation and then explain how the Ensemble Transform Kalman Filter estimates the likelihood.

4.1 MCMC

The pseudo code for the MCMC/ETKF is as follows:

```
function MCMC($y, S, I, R$)
    $chain[0, 0] \leftarrow U(b0min, b0max)$
    $chain[0, 1] \leftarrow U(b1min, b1max)$
    $chain[0, 2] \leftarrow U(mmin, mmax)$
    $chain[0, 3] \leftarrow U(amin, amax)$
    $chain[0, 4] \leftarrow U(gmin, gmax)$
    for $i \in 1000$ do
        $L \leftarrow$findLikelihood($y$, $chain[i]$)
        $chain[i + 1] \leftarrow chain[i] + \epsilon$
        if $L/L_{prev} < U(0, 1)$ then
            $chain[i + 1] \leftarrow chain[i]$
        end if
    end for
    return $chain$    ▷ chain of parameters
end function
```

Markov chain Monte Carlo is an algorithm for sampling from a probability distribution. In particular, this paper uses a Metropolis-Hastings random walk. This algorithm is used when directly sampling from a distribution is challenging and is most useful when the distribution is multi-dimensional. Metropolis-Hastings was selected as the tool for the SIRS model because there are 5 parameters that need to be estimated meaning that the distribution will be 5 dimensional.

For this project, I began by initially drawing each parameter from uniform distributions of varying widths. With the parameter guesses, I found the likelihood given the parameters and began to cycle through the MCMC. This entailed adding some amount of noise (some perturbation) to my previous parameter estimates and then verifying that the new parameter estimates did not leave a certain boundary. The reason for adding the boundary condition is because none of the parameters could be negative for the SIRS model. If the new parameter estimates (the proposal) wander outside of the boundary, then we reject the proposal and keep the old parameter estimates. Otherwise, if the proposal is still within the bounds, we continue to calculate the likelihood given the proposal. Then, comparing the ratio of the new likelihood over the old likelihood to a random uniform from 0 to 1, we determine whether or not to keep the proposal. If the ratio is greater than the random uniform value, meaning that the proposed parameter estimates are more likely to be closer to the true parameters, we accept the proposal. Otherwise, we reject the proposal and keep the previous parameter estimates (because the old values were more likely). This MCMC cycle repeats until a fixed amount of iterations have completed (for example, 1000 iterations) or until the chains appear to have remained stable for a fixed period of time (for example, 200 iterations). In this paper, I chose to use a fixed 1000 iterations.

Some MCMC best practices are determining whether there was a burn in period, thinning the chains to reduce autocorrelation and verifying that the chains do in fact appear to converge to correct parameter estimates using convergence diagnostics. Typically before studying the distribution of the estimated parameters by studying the resulting chain, it is recommended to inspect the chain to see if there is a period of time at the beginning where the chain appears to be wandering around a bit more and remove those sections of the chain. This prevents the distribution of the parameters from being improperly skewed. As for thinning, the primary goal is to reduce autocorrelation. The MCMC's resulting samples are intrinsically correlated because each sample is drawn using the previous sample. With Metropolis-Hastings, we can control autocorrelation slightly by adjusting the variance of the proposal distribution however the underlying issue is not eliminated. The best way to reduce autocorrelation for the MCMC is to increase the amount of lag between samples because typically as lag increases, autocorrelation decreases. The process of removing samples to reduce autocorrelation is often called thinning. The final best practice I will discuss here is the verification of convergence. There are number of metrics that can be used to do so however the primary one used in this paper is the Gelman-Rubin statistic [4]. The diagnostic analyzes convergence by studying the difference between multiple Markov chains. Convergence is evaluated by measuring the between chain and within chain variances for each estimated parameter. If the differences are large, then the chains likely do not converge. If the statistic is close to 1, the chains have most likely converged.

4.2 ETKF

Finally for this section, I will describe the Ensemble Transform Kalman Filter algorithm and how it is used to find the likelihood given parameter estimates. The ETKF is an Ensemble Square Root Filter. The ETKF differs from the EnKF because the EnKF maintains the BLUE (Best Linear Unbiased Estimator)/ Kalman Filter update for the posterior covariance only in the long run. The ETKF generates a posterior mean and a posterior ensemble that exactly satisfy the BLUE/KF formulas:

$$\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_{j+1|j} + \boldsymbol{K}_{j+1}(\boldsymbol{y}_{j+1} - \boldsymbol{H}_{j+1}\boldsymbol{\mu}_{j+1|j})$$
$$\boldsymbol{C}_{j+1} = (\boldsymbol{I} - \boldsymbol{K}_{j+1}\boldsymbol{H}_{j+1})\boldsymbol{B}_{j+1}$$

When the ETKF was originally suggested [5], the posterior covariance was smaller than it should be and the posterior mean was incorrect as well because the perturbation matrix was not a true perturbation matrix (the sum of the columns was not exactly zero). [6] The original paper required the use of a square root matrix which is considered somewhat unnatural. The new paper [7] which eliminates the use of square root matrices is now considered the “correct” ETKF algorithm.

The algorithm for the ETKF applied to the SIRS model is as follows. We begin by creating an ensemble matrix with dimensions $3 \times N$. The ensemble matrix has 3 rows because the SIRS model has 3 state variables (S, I, R) and has N rows because we want to create an ensemble matrix with N ensemble members. The first row (S) was created by drawing N samples from a random normal with mean S_{obs} and variance I_{obs}. The second row (I) was created by subtracting the first row from the population. The third row (R) was set to zeros.

Next, we begin assimilating and forecasting using the ETKF algorithm. We cycle for each observation so our loop is as long as the number of observations that we have. Within each cycle, we find the ensemble mean, construct a perturbation matrix **A** (ensemble member minus the ensemble mean), update the ensemble mean to find the posterior mean and update the perturbation matrix to find the posterior perturbation matrix. Then finally, *for two-thirds of the observations*, we assimilate by updating the ensemble by adding the updated mean to $\sqrt{N-1}$ times the updated perturbation matrix. Because in the SIRS model, every state variable is strictly positive, I needed to add a check for negative numbers in the updated ensemble matrix. If anything was negative, then I shifted the negative values up to a random uniform between 0 and 0.1 (I will discuss the effects of this later in the paper). Then for the two-thirds of the observations that were assimilated *and* for the one-third of observations that were not assimilated, we forecast using the ensemble in the SIRS model (with the proposed parameters).

Finally, after we finish updating our ensemble, we are ready to calculate the likelihood. Here, we choose to use the log-likelihood as the output from the likelihood was quite small. We do so by:

$$L = \frac{1}{N} \sum_{i=1}^{N} e^{\sum_{j=2d/3}^{d} -\frac{1}{2}\left(\frac{obs[j]-i[1]}{\sqrt{R}}\right)^2}$$
$$\ell = log(L)$$

where N is the number of ensemble members, d is the number of observations, $i[1]$ selects the ensemble member corresponding to the number of infected people and R is the observation variance. This log-likelihood is then used to assess whether the proposed parameters should be kept. Because I chose to use the log-likelihood, the MCMC Metropolis-Hastings acceptance criteria changes from the ratio of logs to the difference between logs.

5 Results

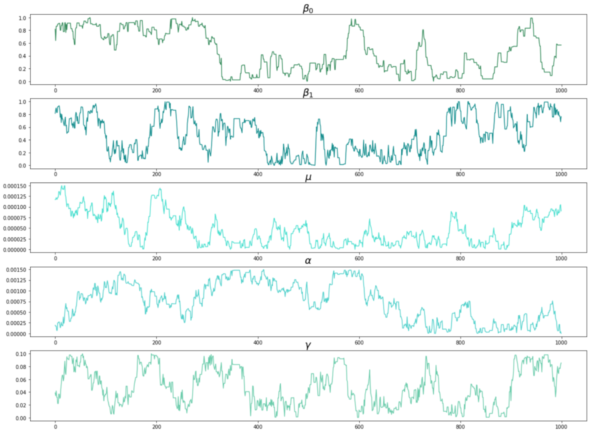
The methodologies discussed in the previous section were applied to two different sets of data. The first was a simple “toy” model and the second dataset was death data from the CDC from 2010-2011 [8].

5.1 Toy Model

To implement the MCMC/ETKF on the toy model, I needed to create the “true” data. To do so, I set the parameters for the SIRS model using the values specified in “Accounting for non-stationarity in epidemiology by embedding time-varying parameters in stochastic models” [2] and generated the true values of S, I and R over time. I generated 365 values for each state variable. S was initialized at 1000, I was initialized at 3 and R at 0.

The observations for the toy model were created from the number of infected people. In particular, the observations were created by adding some perturbations to each term in the list of true infected values. I then ran the MCMC/ETKF algorithm on the observations to try to estimate the parameters for the

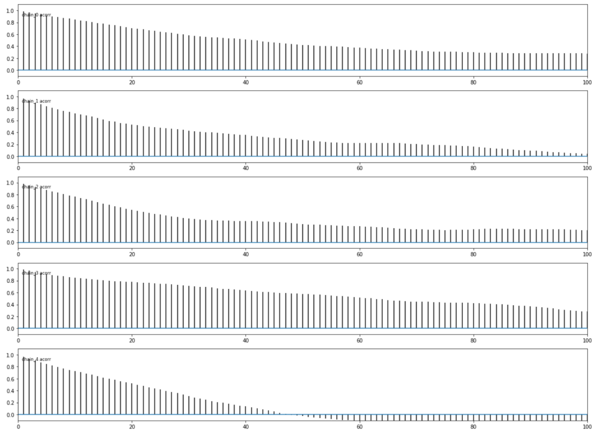
SIRS model. The goal of this model was to compare the true preset parameters to the parameters that the MCMC/ETKF estimated. The following are the chains of the parameters for 1000 iterations of the MCMC. This model needed to be run multiple times until our chain was accepting the proposal approximately 25% of the time.



For the toy model, my initial guesses for the parameter in the MCMC were:

- $\beta_0 \sim$ Uniform(0, 1)
- $\beta_1 \sim$ Uniform(0, 1)
- $\mu \sim$ Uniform(0, 0.0003)
- $\alpha \sim$ Uniform(0, 0.0015)
- $\gamma \sim$ Uniform(0, 0.1)

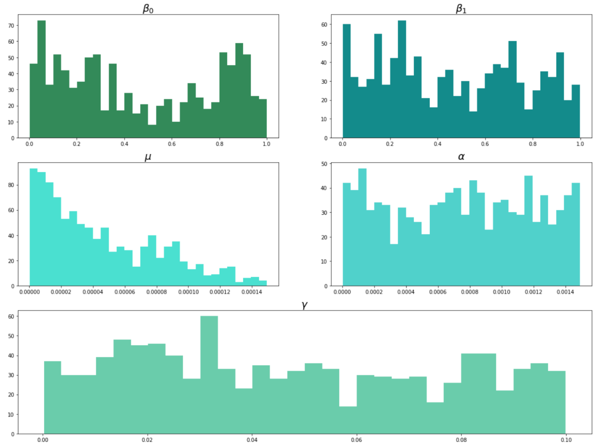
As we can see in the chains above, none of the chains appear to have a burn in period with the exception of β_0. I chose not to remove the burn in period because the chain was short (1000) and the majority of the parameters did not seem to have one.



The figure above is of the autocorrelation between chains. The autocorrelation plot helps determine which thinning parameter should be selected. For the toy model, I chose not to worry about thinning because the estimated parameters appeared to be fairly close to the true parameters already. If the true parameters were not known, then the autocorrelation would matter more.

The Gelman-Rubin [4] statistic for this chain of parameters is 0.89. Because a statistic close to one is considered convergent, we can make the argument that this chain does converge.

The following figure is the estimated distribution of the parameters.



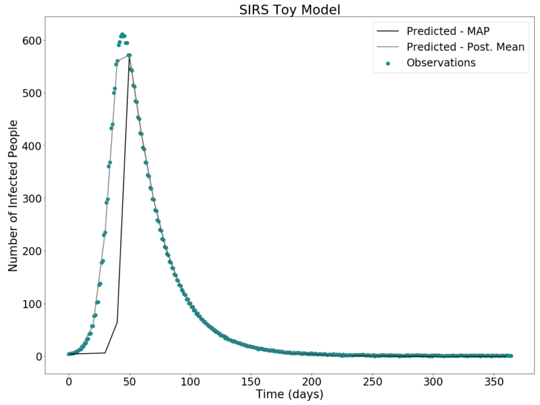
As we can see, the distributions do not appear to be particularly unimodal.

	β_0	β_1	μ
True	0.65	0.4	5.48e-5
Post. Mean	0.468	0.469	4.54e-5
MAP	0.038	0.732	1.48e-5

	α	γ
True	0.00039	0.07143
Post. Mean	0.00075	0.04741
MAP	0.00148	0.08296

The tables above compare the true parameter values to two different estimates. The MAP (maximum a posteriori) takes the mode of each posterior distribution and the posterior mean takes the mean of each posterior distribution. As we can see, the MAP appears to do a worse job than the posterior mean in this case. This is likely because the parameters have multiple modes and the chain does not explore the whole parameter space. The MAP appears to do super poorly on β_0 and the primary reason for that is likely due to the fact that the burn in period was not removed.

To compare the true number of infected to the estimated, we can run the ETKF with the estimated parameters and plot the estimates:

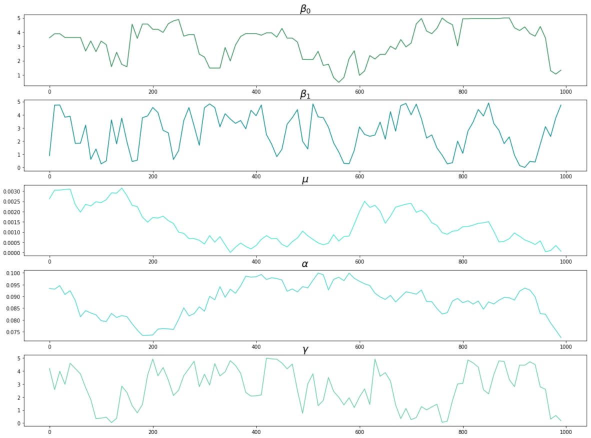


In the figure above, the turquoise points are the observations, the grey line is the estimate using the MAP of the parameters and the black line is the estimate using the posterior mean estimates of the parameters. While the MAP does a fairly good job at estimating the curve, the posterior mean does even better. Because the posterior mean parameter estimates did not match up exactly with the true parameters but the estimated number of infected people matches up fairly well, it's possible that there is no "best" set of parameters for the SIRS model.

5.2 CDC Death Data

Moving to a real dataset, I chose to use CDC death data from the 2010-2011 flu season in Colorado. Because the number of deaths does not necessarily correspond 1 to 1 with the number of infected people, I needed to extrapolate to estimate the number of infected people. I accomplished that by multiplying the quantity of deceased by $21000/(37*1000)$ where $21000/37$ represents the ratio of number of infected to number of deceased in the 2010-2011 flu season [9] and the value is divided by 1000 to scale the data. I also retrieved the population of Colorado in 2010 and 2011 and averaged that, subtracted the number of infected people, and divided by 1000 to get the scaled number of susceptible. The number of recovered people was set to zero again.

Now, similar to the Toy model, I ran this data through the MCMC/ETKF algorithm to estimate the parameters for the model. I ran it for 1000 MCMC iterations and found the following chains:

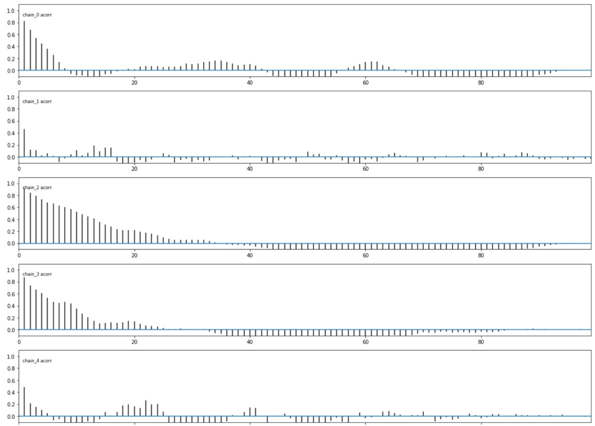


For the model with death data, the MCMC initial guesses were:

- $\beta_0 \sim$ Uniform(0, 5)
- $\beta_1 \sim$ Uniform(0, 5)
- $\mu \sim$ Uniform(0, 0.01)
- $\alpha \sim$ Uniform(0, 0.1)
- $\gamma \sim$ Uniform(0, 5)

Like to toy model, this model needed to be run multiple times until our chain was accepting the proposal approximately 25% of the time. Notice that these initial guesses are coming from a wider uniform distribution. The reason for this is because the true parameters are unknown and thus a wider initial guess is better. I vaguely tried to initialize the parameters around the parameters mentioned in Cazelles et. al. however those were the best parameters for a different dataset which lead me to specifying a wider initial distribution.

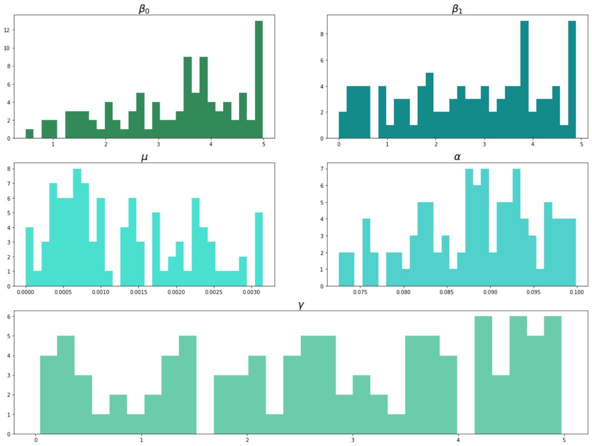
After finding a thinning lag of 10, the autocorrelation plot looks much better:



The lower autocorrelation, the higher efficiency we have in the chain and the better our estimates are.

The Gelman-Rubin statistic for the chain of parameters is closer to 0.88 meaning that the chains converge slightly more poorly than the chains for the toy model. However, 0.88 is still quite close to 1.0 meaning that chain does converge.

The following figure is the estimated distribution of the parameters derived from the chains *after* thinning:



Once again, by observing the plot above we can't particularly see a single value that the distribution seems to be centered at. This implies that the posterior distributions are likely quite wide. In the case of the deaths data, we have a wide prior and a wide posterior distribution.

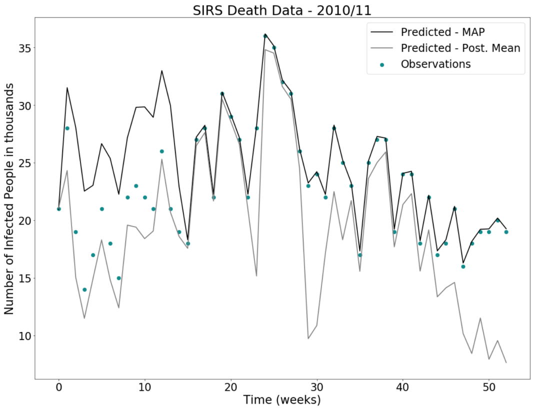
The parameter estimates that we can use to plot our estimated number of infected people are the MAP and the posterior distribution:

	β_0	β_1	μ
Post. Mean	3.33	2.69	0.0013
MAP	4.95	0.611	0.0008

	α	γ
Post. Mean	0.088	2.76
MAP	0.076	0.38

The majority of the estimates (β_0, μ, α) appear to be of the same magnitude between the MAP and the posterior mean. However, β_1 and γ appear to be quite a bit smaller in the MAP than the posterior mean. In the case of the death data, it is likely that the MAP would be the better estimate because the parameters appear to have multiple modes in which case the posterior mean is typically a poor estimate.

Once again, I ran the ETKF with the estimated parameters (MAP and posterior mean) and plotted the results to compare them with the true data:



Like the toy model, the black line is the estimate using the MAP parameter estimates, the grey line is using the posterior mean, and the turquoise points are the observations (in this case, simply the extrapolated number of infected people).

Remember in the toy model, the posterior mean was able to find the true parameters slightly better than the MAP. When using real data (at least in this case), the MAP appears to find the correct number of infected people better. It's interesting because the MAP and the posterior mean appear to do equally well at the beginning. Arguably, the mean almost does better than the mode. About halfway through the flu season, the MAP estimate appears to start clinging to the observations. While this may look like overfitting, what is actually happening is that in the ETKF, the estimates start trusting the observations more and not using the ensemble as much. The posterior mean drifts off a bit in the middle and at the end and that is likely an indication that the posterior mean is just not a great estimator in this case.

6 Discussion

This paper touched on a number of challenges that can arise in a study like this one. For example, a big challenge is selecting a SIRS model. There are multiple different implementations of the dynamic system. For this paper, I selected the SIRS model that Cazelles et. al. used in their 2018 paper.

Another challenge that arises in any parameter estimation problem is knowing whether or not the parameters are "correct". There is no way to know whether the parameters are exactly correct but plotting the observations and studying the results with the estimated parameters are a decent way to determine how well the estimated parameters do.

Any use of the MCMC comes with its own set of challenges. For example determining where to start your chain (choosing a prior), tuning the acceptance rate — default 25% — by tweaking the random walk covariance and evaluating convergence. Some methodology for the later was discussed in the MCMC section of this paper. Choosing a prior can be done by selecting a tighter prior if you are fairly confident of the general location of the parameters. If you have no inclination as to where the parameters may be, a wider prior should be selected with bounds that seem reasonable given the problem being solved. Selecting the best random walk covariance, for this paper, was done via trial and error until a 25% acceptance rate was achieved.

The model with the CDC data was also susceptible to representation error. The CDC data contained number of deaths for pneumonia and influenza. There is error extrapolating the number of infected people because the estimation was rather simple (just multiplied the number of deceased by a constant) and because it may have been an overestimate due to the inclusion of pneumonia.

Additional challenges included the intrinsic randomness of the MCMC/ETKF model and the necessity of positivity in the SIRS model. The assumption of the SIRS model is that the population is constant and that the number of people in each group is positive. During the ETKF ensemble update, the number of recovered people would sometimes become negative and I needed to push those values up to a small positive real number. However, by doing this, I ended up modifying the population and causing it to be larger than it was originally. To mitigate this issue, if $S + I$ was greater than the population, I reset the susceptible to the population minus 21 (the average number of deaths throughout the season) and the number of infected to 21.

7 Conclusion

In conclusion, it is clear that the ETKF does a decent job at estimating parameters for the SIRS model however due to the Gaussian assumption, a different model may have been better suited. In the future, I would like to implement this same model but with a particle filter (removes Gaussian assumption) and eliminate the “hacky” fix of maintaining a constant population and keeping the state variables positive. One option for doing so would be to simply not keep track of the R variable however its possible this does not entirely solve the problem.

References

[1] Leon Danon, Ashley P Ford, Thomas House, et al. “Networks and the epidemiology of infectious disease”. In: *Interdisciplinary perspectives on infectious diseases* 2011 (2011).

[2] Bernard Cazelles, Clara Champagne, and Joseph Dureau. “Accounting for non-stationarity in epidemiology by embedding time-varying parameters in stochastic models”. In: *PLoS computational biology* 14.8 (2018), e1006211.

[3] Wan Yang, Alicia Karspeck, and Jeffrey Shaman. “Comparison of filtering methods for the modeling and retrospective forecasting of influenza epidemics”. In: *PLoS computational biology* 10.4 (2014), e1003583.

[4] Andrew Gelman, Donald B Rubin, et al. “Inference from iterative simulation using multiple sequences”. In: *Statistical science* 7.4 (1992), pp. 457–472.

[5] Craig H Bishop, Brian J Etherton, and Sharanya J Majumdar. “Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects”. In: *Monthly weather review* 129.3 (2001), pp. 420–436.

[6] Ian Grooms. *Lecture notes in Data Assimilation CU Boulder 2019*. Oct. 2019.

[7] Xuguang Wang, Craig H Bishop, and Simon J Julier. “Which is better, an ensemble of positive–negative pairs or a centered spherical simplex ensemble?” In: *Monthly Weather Review* 132.7 (2004), pp. 1590–1605.

[8] *Deaths from Pneumonia and Influenza (P & I) and all deaths, by state and region.* https://data.cdc.gov/Health-Statistics/Deaths-from-Pneumonia-and-Influenza-P-I-and-all-de/pp7x-dyj2. Accessed: 2019-09-10.

[9] *Disease Burden of Influenza.* https://www.cdc.gov/flu/about/burden/index.html. Accessed: 2019-12-04.