Distribution Agreement

Signature:

Hung Tan Ngo March 29, 2024

Parameter Tuning for SIR-Related Models With Variational and Bayesian Methods

By

Hung Tan Ngo

Alessandro Veneziani, Ph.D.
Advisor

Mathematics

Alessandro Veneziani, Ph.D.
Advisor

Elizabeth Newman, Ph.D.
Committee Member

Talea L. Mayo, Ph.D.
Committee Member

2024

Parameter Tuning for SIR-Related Models With Variational and Bayesian Methods

By

Hung Tan Ngo

Alessandro Veneziani, Ph.D.
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Mathematics

2024

Abstract

Parameter Tuning for SIR-Related Models With Variational and Bayesian Methods
By Hung Tan Ngo

The event of COVID-19 has put many mathematical models in competition to capture the disease's dynamic. The effectiveness of such a model is only possible with a process to tune its parameters for the available dataset. This thesis presents two methodologies — Trust Region, a variational approach, and Ensemble Kalman Filter (EnKF), a Bayesian approach — to solve the above issue. This project deals with the classical Susceptible, Infectious, and Recovered (SIR) epidemiology model and its variants. We compare the efficiency of our approaches through three SIR-related models: Epidemic SIR, Endemic SIR, and SIRW. Firstly, employing the variational method, especially the trust region method, and the PyBOBYQA algorithm, we fine-tune our models' parameters under noise-free and noise-inclusive conditions. Similarly, we utilize the Ensemble Kalman Filter method to explore the optimal sets for our models when white noise is presented and not presented. The results show that the Trust Region method performs well with the two basic SIR models under every condition, but this approach is not capable of handling more sophisticated models like SIRW. EnKF shows potential findings across the three models when the dataset is absent of noise. However, when a mild amount of noise is introduced, our optimization only shows success for the epidemic SIR and SIRW cases. With highly random datasets, we can only tune the correct parameters for the epidemic SIR model. This project serves as a first step in finding efficient optimization methodologies for non-linear models under different conditions.

Parameter Tuning for SIR-Related Models With Variational and Bayesian Methods

By

Hung Tan Ngo

Alessandro Veneziani, Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Mathematics

2024

Acknowledgments

I want to express my greatest gratitude to Professor Veneziani for his invaluable help throughout the development of my honors thesis. Since my sophomore year, I have always been grateful for his guidance and mentorship. Most of my knowledge in research has been accumulated during my time working with him. His passion for mathematical applications kindled my interest in parameter optimizations and forecasting, and his profound understanding of the subject showcased me a new possibility in mathematics.

Additionally, I must send my deepest appreciation to all my family members and friends who have always supported and encouraged me along my entire undergraduate journey. Their presence has been motivating and has provided solace during challenging times.

My research would have not been possible without the help of these remarkable individuals. I am deeply thankful for everyone who has been by my side, for your support has been indispensable. Thank you for being a part of this journey.

Contents

List of Figures

Chapter 1

Introduction

In recent years, we have witnessed unprecedented public health challenges that emphasized the role of contagious disease modeling in understanding and managing epidemics. The Susceptible-Infected-Recovered (SIR) model and its variants are potential tools to provide insights into future outbreaks. This will be extremely helpful in facilitating strategic public health interventions to minimize the deadly impacts as seen in the case of COVID-19. To extract the maximal values out of these models, researchers need to tune their parameters to accurately capture the disease's dynamic. In this paper, we approach this topic in two different ways: deterministic and probabilistic. In chapter 3, we will use the variational method to tune our three studied models, epidemic, endemic SIR, and SIRW models. Then, we will compare our result with the Bayesian approach discussed in chapter 4 to see the effectiveness of each method. Moreover, we will also be testing our models under noise-free and noisy conditions in an attempt to mimic real-world cases.

The goal of this work is to offer practical tools for parameter tuning that can be helpful to epidemiologists and public health officials in their ongoing battle against infectious diseases. In chapter 5, we will be talking about our findings and potential future work.

Chapter 2

Introduction to SIR-Related Models

2.1 Epidemic SIR Model

The infamous SIR model is widely used in predicting species' populations and disease spreading due to its effectiveness and simplicity. In the basic epidemic SIR model, we have three compartments interacting with each other: Susceptible $S(t)$, Infected $I(t)$, and Recovered $R(t)$. They solve the following systems of ODEs in time:

$$\begin{cases} d_t S = -\beta IS/N \\ d_t I = \beta IS/N - \gamma I \\ d_t R = \gamma I \end{cases} \tag{2.1}$$

where β is the effective contact rate, which represents the average rate at which infected individuals are in contact with susceptible individuals and transmit the disease, and γ is the average recovery rate, which represents the rate at which infected individuals will recover or die, moving from group $I(t)$ to group $R(t)$.

2.2 Endemic SIR Model

The endemic SIR model incorporates the demography of the populations through a new parameter μ, which is the reproduction/death rate of the population. We assume this rate to be constant and project that our population does not vary over time. This specific model aims for long-term predictions of the disease.

$$\begin{cases} d_t S = \mu N - \mu S - \beta IS/N \\ d_t I = \beta IS/N - \gamma I - \mu I \\ d_t R = \gamma I - \mu R \end{cases} \tag{2.2}$$

A more complex scenario will be discussed below, but to preserve the main purpose of our research– to accurately tune models' parameters for presented datasets, we prefer working with simple models first before moving on to more sophisticated systems.

2.3 SIRW Model

Long-term forecasts cannot be achieved simply through the research of the interactions among the three compartments, the Susceptible (S), Infectious (I), and Recovered (R). The SIRW model deals with a new group called the Waning Immunity (W). This new compartment represents a more realistic scenario where immunity after infection (or vaccination) wanes over time. This group consists of individuals who move out of the Recovered compartment and will move back to the Susceptible compartment due to the waning of immunity.

$$\begin{cases} d_t S = -\beta IS/N + \chi W \\ d_t I = \beta IS/N - \chi I - \gamma I \\ d_t R = \gamma I - \omega R + \alpha IW/N \\ d_t W = \omega R - \alpha IW/N - \chi W \end{cases} \tag{2.3}$$

In this system, β and γ are identical to the other two SIR models. ω controls how fast recovered individuals (R) lose their immunity. The term $\frac{IW}{N}$ describes the immunity booster action due to the contact between an infected individual and a waning one. α is the efficiency of the boosting action.

Chapter 3

Deterministic Approach

Our SIR-related models are nonlinear, and computing exact numerical solutions for every new dataset received is not feasible and practical. Therefore, we turn to the variational approach to achieve approximated solutions through the minimization of a loss function we will later define.

3.1 Variational Method

We take the case of the epidemic SIR model with three main components — $S(t)$, $I(t)$, and $R(t)$ — and two parameters — γ, β.

$$\begin{cases} d_tS = -\beta IS/N \\ d_tI = \beta IS/N - \gamma I \\ d_tR = \gamma I \end{cases} \tag{3.1}$$

For this system, our goal is to explore the optimal set of γ and β that best fits our system to the presented dataset. We define a loss function, which is our objective for minimization.

$$J = \frac{1}{2}(|S_{data}(t_i) - S_{computed}(t_i)| + |I_{data}(t_i) - I_{computed}(t_i)| + |R_{data}(t_i) - R_{computed}(t_i)|) \tag{3.2}$$

We ensure the effectiveness of our optimization by minimizing J, and the smaller J means the more accurate our computed, or predicted, values of S, I, and R. Our choice of to measure the absolute value error instead of the squared error sparkled from our aim for simplicity. We note that this measure of J caused the function to be less regular, but this did not affect our result as our method introduced later is derivative-free. Our idea is to find β and γ, such that

$$\beta, \gamma = \underset{\beta,\gamma}{\operatorname{argmin}}\, J(S(\beta,\gamma), I(\beta,\gamma), R(\beta,\gamma)) = \underset{\beta,\gamma}{\operatorname{argmin}}\, J(\beta,\gamma) \tag{3.3}$$

A common approach to such problems is to find the solution through the use of the function's gradient, we set

$$\nabla_{\beta,\gamma} J = 0 \tag{3.4}$$

Although this methodology can be highly accurate, it is computationally heavy and time-consuming. Solving this problem requires the so-called "adjoint" problem, which is backward in time. So we need to solve SIR and its adjoint several times over the entire time domain. Especially, when dealing with a complex system and the need to perform the calculation daily, we will stumble into an ineffective process. For this reason, we look into the trust region method that does not require the derivation process.

3.2 Trust Region (Derivative-Free)

Trust Region methods can be applied to solve complex systems as they do not require expensive computation power. The essential characteristic of these methods that contributes to this convenience is their derivative-free nature. We will use the Py-BOBYQA algorithm [4], a built-in Python library, to solve our systems. Py-BOBYQA is a model-based derivative-free optimization method where a local model for the objective is constructed by interpolation and minimized on each iteration, and, more specifically, it has its roots in Powell's BOBYQA [7]. In this method, we construct an interpolation-based model and minimize it on each iteration. At each iteration k, we have a collection of points Y_k, where $|Y_k| \in \{n+1, \ldots, (n+1)(n+2)/2\}$, and we construct a local model for the objective

$$f(\mathbf{x}_k + \mathbf{s}) \approx m_k(\mathbf{s}) = c_k + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top H_k \mathbf{s} \tag{3.5}$$

satisfying the interpolation conditions

$$m_k(\mathbf{y}_t - \mathbf{x}_k) = f(\mathbf{y}_t), \text{ for all } \mathbf{y}_t \in Y_k. \tag{3.6}$$

If $|Y_k| < (n+1)(n+2)/2$, the solution to (3.6) is non-unique. We use the remaining degrees of freedom by solving

$$\min_{c_k, \mathbf{g}_k, H_k} \|H_k - H_{k-1}\|_F^2 \quad \text{subject to (3.6)}, \tag{3.7}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm and the convention $H_{-1} = 0$. The value of $|Y_k|$ is input by us, and the larger it is the more objective information is captured. For smooth problems, the default value $|Y_k| = 2n+1$ is used, but for noisy problems, we use the default value $|Y_k| = (n+1)(n+2)/2$ [3].

The next steps are done using a trust-region method[6] to ensure global convergence. We keep a parameter $\Delta_k > 0$ and calculate a new step by approximately solving

$$\mathbf{s}_k \in \operatorname*{argmin}_{\|\mathbf{s}\| \leq \Delta_k} m_k(\mathbf{s}) \tag{3.8}$$

The solution to (3.8) can be found elsewhere (e.g.[5]) so we will not go into detail of this matter. We continue to evaluate if our function is minimized to our favor by looking at the objective at $\mathbf{x}_k + \mathbf{s}_k$ through

$$r_k = \frac{\text{actual decrease}}{\text{expected decrease}} := \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{0}) - m_k(\mathbf{s}_k)} \tag{3.9}$$

If r_k is sufficiently large, we accept the step ($\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$) and increase Δ_k, if not we decrease Δ_k and reject the step ($\mathbf{x}_{k+1} = \mathbf{x}_k$).

The following section is where we apply the PyBOBYQA library to optimize parameters for our models. More details or explanation behind the operation or mathematical background can be found in the two papers, [4] and [3].

3.3 Variational Procedure With PyBOBYQA

3.3.1 Epidemic SIR Model

For the SIR model, We tested the consistency of our deterministic approach by running our optimization on the dataset generated by the same SIR model. To explore the performance of the method under noisy conditions mimicking reality, we introduce noise later in our test.

Without Noise

Our dataset is the epidemic SIR model in (2.1) with $\beta = 0.15$ and $\gamma = 0.06$, and we input our initial guess for the parameters to be

$$\beta = 0.01,\ \gamma = 0.01, \tag{3.10}$$

and bounds

$$\beta \in [0.0, 1.0],\ \gamma \in [0.0, 0.3] \tag{3.11}$$

With the same model, the epidemic SIR, to capture the behavior of this dataset, we have our result to be

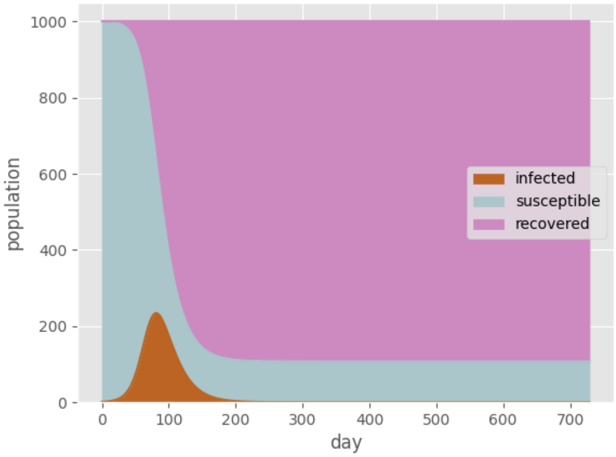

Figure 3.1: Simulation result of the epidemic SIR model

Our optimization returns the value for the parameters to be

$$\beta = 0.15,\, \gamma = 0.06000001, \tag{3.12}$$

which are highly accurate, given our data parameters, $\beta = 0.15$ and $\gamma = 0.06$.

This is just a consistency test on the data that is generated by the same model we assume for our analysis and is also noise-free.

With Noise

In reality, it is rarely the case where the data presents its true nature without any white noise. Therefore, we wish to test our approach under the circumstance where noise is available. We add δ_S, δ_I, and δ_R such that

$$\delta_S,\, \delta_I,\, \delta_R \in \mathcal{N}(0, 50.) \tag{3.13}$$

into our system's components, $S(t)$, $I(t)$, and $R(t)$. The amount of noise introduced into the system is high, and we first turn off the "objfun_has_noise" feature from PyBOBYQA to see the impact of noise to our optimization.

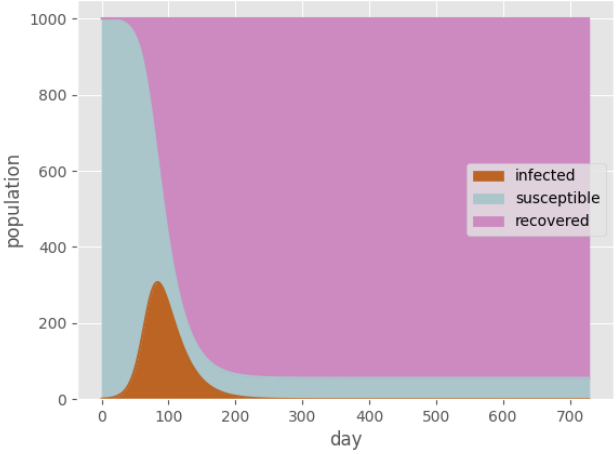

Figure 3.2: Simulation result of the epidemic SIR model under noisy data and without noise detection feature

Our optimized parameters are

$$\beta = 0.13691506,\ \gamma = 0.04482117 \tag{3.14}$$

We can visibly see the large errors accounting by the introduce of noise to our system. Now, let's see how these errors will change if the "objfun_has_noise" feature is on.

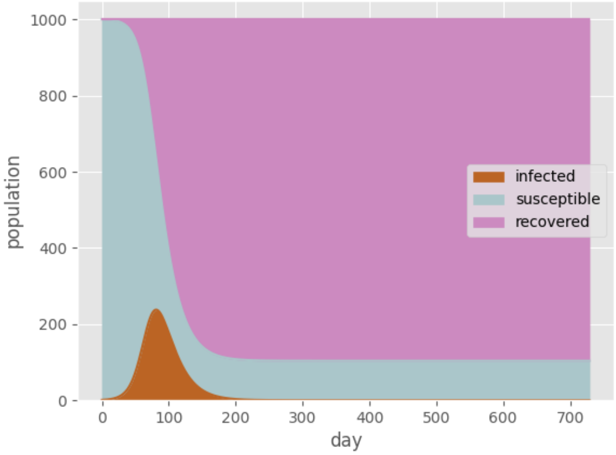

Figure 3.3: Simulation result of the epidemic SIR model under noisy data and with noise detection feature

Our predicted parameters to be

$$\beta = 0.14960063,\ \gamma = 0.05920766 \tag{3.15}$$

The result is improved and captures the true distribution of our generated data. We need to analyze now what happens with more complex models.

3.3.2 Endemic SIR Model

Moving to the endemic SIR model, we have another parameter, μ, to optimize. We will slowly increase the complexity of our system to see the capability of this trust region method in the PyBOBYQA library.

Without Noise

We generated our dataset with the endemic SIR model in (2.2) where our parameters β, γ, and μ are 0.15, 0.06, and 0.004 respectively. We have our initial or ”guessing” values of our parameters

$$\beta = 0.01,\ \gamma = 0.01,\ \mu = 0.001 \tag{3.16}$$

with bounds

$$\beta \in [0.0, 1.0],\ \gamma \in [0.0, 0.3],\ \mu \in [0.0, 0.5] \tag{3.17}$$

Using the endemic SIR model as a predicting model, our optimization process generated the following result for the optimal set of parameters.

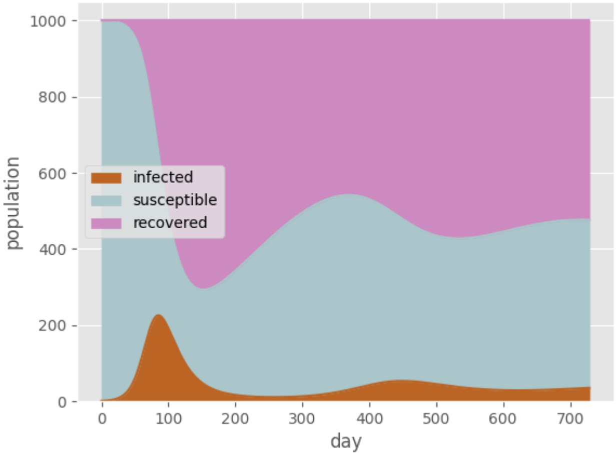

Figure 3.4: Simulation result of the endemic SIR model

The estimate values of our parameters for the above simulation are

$$\beta = 0.015000002,\ \gamma = 0.05999938,\ \mu = 0.00400031 \tag{3.18}$$

This optimized set of parameters approximately equal to the actual values of parameters in the generated dataset, and the result is proved to be consistent across our numerous trials.

With Noise

Next step, we gain insights into the practicality of our approach through the introduction of white noise into the dataset. We make our data noisy by adding δ_S, δ_I, and δ_R such that

$$\delta_S,\ \delta_I,\ \delta_R \in \mathcal{N}(0, 50.) \tag{3.19}$$

This makes our dataset very noisy as our whole population consists of only 1000 people. As we have seen from the epidemic SIR case, a high randomness in our dataset did not have any significant effect on our optimization. We hope to explore this aspect in the endemic SIR as well, since the endemic SIR model is more complex than the epidemic one — the adding of one new parameter μ. We also test out the role of "objfun_has_noise" feature of the PyBOBYQA function. Firstly, we turn off this feature and see how our predicted parameters behave

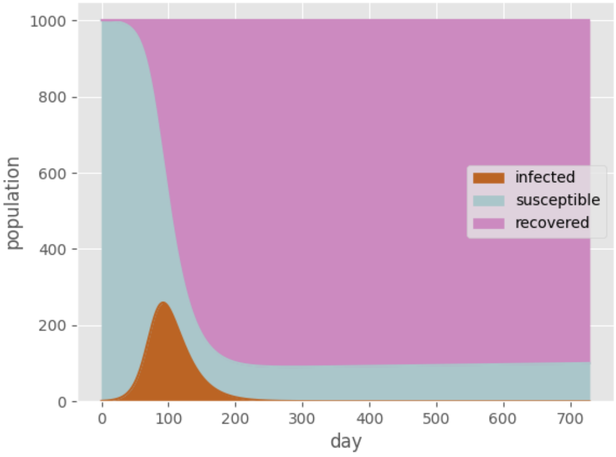

Figure 3.5: Simulation result of the endemic SIR model under noisy data and without noise detection feature

The corresponding values for the above distribution's parameters are

$$\beta = 0.129127966,\ \gamma = 0.0482208523,\ \mu = 0.0000257470483 \tag{3.20}$$

Here, we can see that the errors for β and γ are consideratly bigger than previous attempt without noise, but the observed error for the generated μ is concerning when our predicted μ is only 0.5% of the actual value.

Now, we turn on the "objfun_has_noise" feature

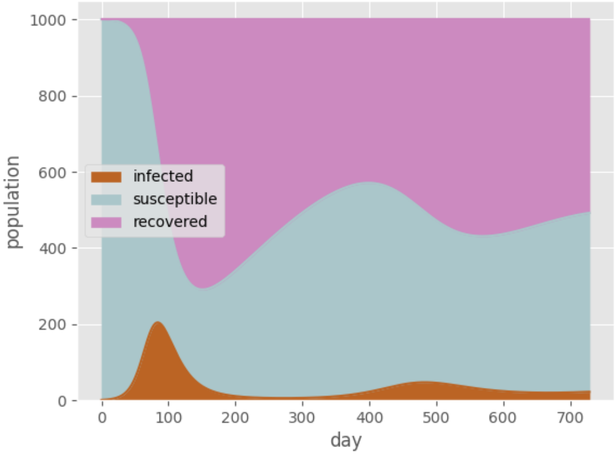

Figure 3.6: Simulation result of the endemic SIR model under noisy data and with noise detection feature

And the corresponding values for the above distribution's parameters are

$$\beta = 0.14675407,\ \gamma = 0.05483139,\ \mu = 0.00483611 \tag{3.21}$$

This result is much improved, and given that our data is so noisy, our approach has done a good job of isolating the noisy part.

3.3.3 SIRW Model

The SIRW model has another compartment $W(t)$, so our new objective function J will be alerted to

$$J = \frac{1}{2}(|S_{data}(t_i) - S_{computed}(t_i)| + |I_{data}(t_i) - I_{computed}(t_i)| \\ + |R_{data}(t_i) - R_{computed}(t_i)| + |W_{data}(t_i) - W_{computed}(t_i)|) \tag{3.22}$$

Similar to before, we will first proceed to generate our data with a SIRW model, where $\beta = 0.5$, $\chi = 4$, $\gamma = 0.2$, $\omega = 10$, and $\alpha = 50$. Our initial "guest" for our parameters will be

$$\beta = 0.1,\ \chi = 1.0,\ \gamma = 0.1,\ \omega = 1,\ \text{ and } \alpha = 1, \tag{3.23}$$

and the bounding conditions are

$$\beta \in [0.0, 0.5],\ \chi \in [0.0, 10.0],\ \gamma \in [0.0, 0.5],\ \omega \in [0.0, 20.0],\ \alpha \in [0.0, 100.0] \tag{3.24}$$

This gives us the following result

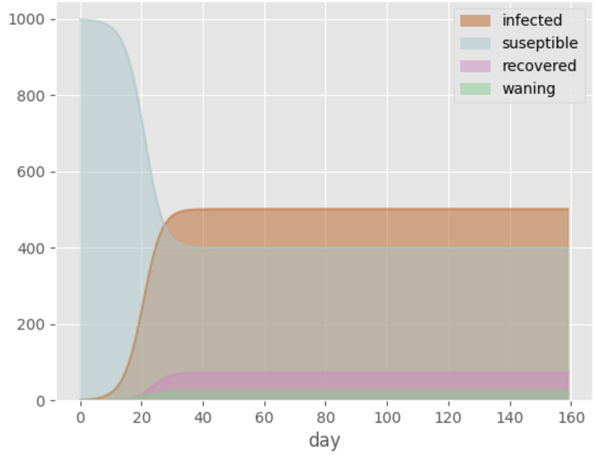

Figure 3.7: Simulation result of the endemic SIRW model

with the set of parameters

$$\beta = 0.49966081,\ \chi = 3.53983071,\ \gamma = 0.19938962,\ \omega = 3.37446941, \text{ and } \alpha = 9.92747379, \tag{3.25}$$

while the predicted β and γ are promising, the rest of the parameters indicate the existence of significant errors. We have foreseen this outcome since the SIRW model is more sophisticated and the trust region method has troubles to accurately approximate its true solution. Taking this in mind and acknowledging the availability of data, we turn our approach into a probabilistic one that will be discussed in the next chapter.

3.3.4 Mixed Models

In this section, we will not use one model to both generate and capture the dataset. We want to explore the two cases: first, when our predicting model is more capable to capture the data behavior itself, and second, when our predicting model is unable to fully model the data.

Epidemic SIR Model as The Data Generator

We generate our data with the epidemic SIR model with $\beta = 0.15$ and $\gamma = 0.05$, then we use the endemic SIR model to explore insights about this dataset. Keeping all bounds and initial guesses identical to those in 3.3.2. We have our result to be:

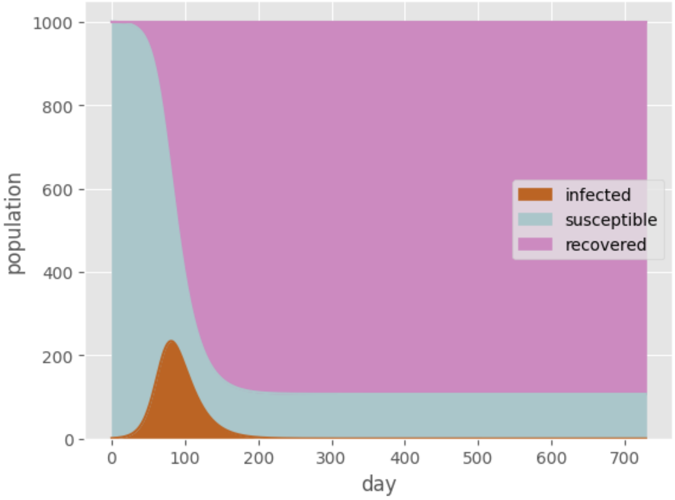

Figure 3.8: Simulation result of the endemic SIR model with the epidemic SIR model as a data generator

and the estimated parameters turn out to be:

$$\beta = 0.15,\ \gamma = 0.05000001,\ \mu = 0. \tag{3.26}$$

We can see that our endemic SIR model perfectly captured the distribution of the data.

Endemic SIR Model as The Data Generator

This case is more common in reality where there are infinite unobserved factors. We generate our data with the endemic SIR model with $\beta = 0.15$, $\gamma = 0.05$, and $\mu = 0.004$. Next step, we use the epidemic SIR model as the predicting tool with all bounds and initial guesses identical to those in 3.3.1.

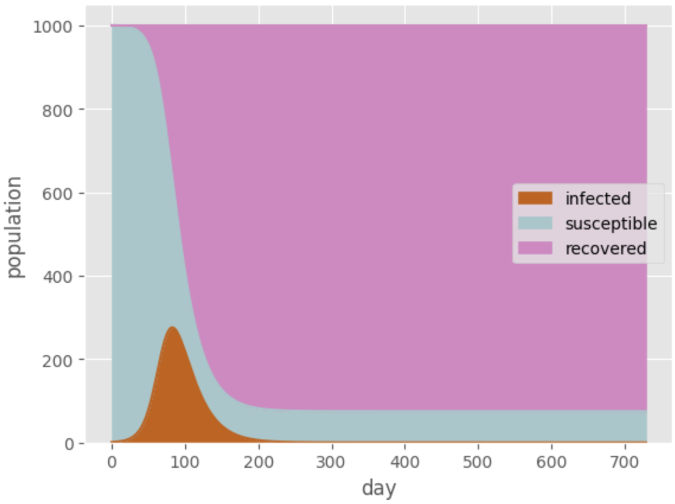

Figure 3.9: Simulation result of the epidemic SIR model with the endemic SIR model as a data generator

our estimated parameters are:

$$\beta = 0.0.14163136, \gamma = 0.050564961 \tag{3.27}$$

The epidemic SIR model somewhat forecasted the behavior of the endemic SIR model, and it is normal to have our estimated β and γ different than the actual values, since our predicting model lacks the sophistication to model the movement determined by μ.

Chapter 4

Bayesian Approach

With a probabilistic approach, we update our parameter estimation based on the new recorded data points. Using this methodology, we overcome the expensive computation required for complex, nonlinear systems. Taking into account new data, we also hope to capture the true nature of datasets generated by the SIR, SIRE, and SIRW models.

4.1 Kalman Filter Method

One of the probabilistic approaches is the Kalman Filter Method. We will first look at the foundational idea behind the KF method for linear problems, then we will extend our scope to nonlinear systems with the use of the Ensemble Kalman Filter (EnKF). We acquire our knowledge about the topic through the two books [1] and [2]. More details about each step and mathematical explanations can be found in there.

For KF, we have two main states: prediction and correction. Our goal is to find the optimal set of parameters for the systems we believe capture the true nature of the phenomenon, COVID-19 in this case. For the prediction step, we utilize the tuned set of parameters that best describe the historical data to forecast the value of the next observation. After the revelation of the actual value for our predicted observation, we account for this disparity, or error, by re-tuning our set of parameters accordingly. We now walk into the details of these two steps.

Prediction Step

At the time index, k.

$$u^{(k)} = A_{k-1}u^{(k-1)} + b^{(k-1)} \tag{4.1}$$

where $b^{(k-1)}$ is a Gaussian white noise in time representing the model error, i.e., $b^{(k)} \sim \gamma(0, Q_k)$, and the errors are not correlated in time, i.e.,

$$\xi(b^{(k)}b^{(l),T}) = Q_k\delta_{kl} \tag{4.2}$$

Here δ_{kl} is the Kronecker delta (it equals 1 if $k = l$ and 0 elsewhere). The measurement process is denoted by

$$z^{(k)} = H_k u^{(k)} + v^{(k)} \tag{4.3}$$

Where $v^{(\cdot)}$ is a Gaussian white noise with variance matrix R_k and assumed uncorrelated with $b^{(\cdot)}$.

What we have is the true state, $u^{(\cdot)}$, and we will first perform our predicting with the deterministic estimate based on the model

$$u_p^{(k)} = A_{k-1}u_*^{(k-1)} \tag{4.4}$$

Where $u_*^{(k-1)}$ is the "true" state $u^{(k-1)}$.

The next step, the correction step, is where it involves the adjustment for the error between the predicted and actual value.

Correction Step

$$u_c^{(k)} = L_k u_p^{(k)} + K_k z^{(k)} \tag{4.5}$$

Estimation error:

$$\begin{aligned} e_p^{(k)} &= u_p^{(k)} - u^{(k)} \\ e_c^{(k)} &= u_c^{(k)} - u^{(k)} \end{aligned} \tag{4.6}$$

By noticing that $e_p^{(k)} = -b^{(k-1)}$ by construction and $\xi(e_c^{(k)}) = 0$ by unbiased correction, we can drive that

$$u_c^{(k)} = u_p^{(k)} + K_k(z^{(k)} - H_k u_p^{(k)}) \tag{4.7}$$

where $z^{(k)} - H_k u_p^{(k)}$ is the innovation, i.e., what is new in $z^{(k)}$ and that is not predictable by $u_p^{(k)}$, and K_k is called the gain matrix, since it weighs the improvement brought to the deterministic estimate by the measures.

As we do not know the true state, so we replace $u_*^{(k-1)}$ with $u_c^{(k)}$ (best estimation)

$$u_p^{(k)} = A_{k-1} u_c^{(k-1)} \tag{4.8}$$

With this equation, we can have

$$e_p^{(k)} = u_p^{(k)} - u^{(k)} = A_{k-1} u_c^{(k-1)} - u^{(k)} = A_{k-1}(u_c^{(k-1)} - u^{(k-1)}) - b^{(k-1)} = A_{k-1} e_c^{(k-1)} - b^{(k-1)} \tag{4.9}$$

As our research only concerned with one-step prediction, the variance matrix of $e_p^{(k)}$ and $e_c^{(k)}$ are

$$\begin{aligned} \Lambda_p^{(k)} &= \xi(e_p^{(k)} e_p^{(k,T)}) \\ \Lambda_c^{(k)} &= \xi(e_c^{(k)} e_c^{(k,T)}) \end{aligned} \tag{4.10}$$

From (4.9), we can derive that

$$e_p^{(k)} e_p^{(k),T} = A_{k-1} e_c^{(k-1)} e_c^{(k-1),T} A_{k-1}^T + b^{(k-1)} b^{(k-1),T} + A_{k-1} e_c^{(k-1)} b^{(k-1),T} + b^{(k-1)} e_c^{(k-1),T} A_{k-1}^T \tag{4.11}$$

Since $b^{(k-1)}$ has no correlation with $e_c^{(k-1)}$, this leads to

$$\Lambda_p^{(k)} = A_{k-1}\Lambda_c^{(k-1)}A_{k-1}^T + Q_{k-1} \tag{4.12}$$

From the above equation, we can derive the Joseph formula.

$$\Lambda_c^{(k)} = (I - K_k H_k)\Lambda_p^{(k)}(I - K_k H_k)^T + K_k R_k K_k^T \tag{4.13}$$

This leads to our Kalman gain matrix

$$K_k = \Lambda_p^{(k)} H_k^T (H_k \Lambda_p^{(k)} H_k^T + R_k)^{-1} \tag{4.14}$$

4.1.1 Ensemble Kalman Filter

The goal of the research was to deal with nonlinear epidemic systems such as SIR, SIRE, and SIRW models, and the linear Kalman Filter method is not a feasible option. Therefore, the focus shifts to the Ensemble Kalman Filter(EnKF) method that while maintaining the beauty of the Kalman Filter method can solve nonlinear systems.

EnKF integrates observational data with models to estimate the state of a dynamic system and is designed to handle non-linear systems more effectively by using a Monte Carlo approach to represent the probability distributions of state estimates [1].

Initialization

We start with an ensemble of state estimates that can be generated by adding random perturbations to the initial condition.

$$x_{i,0}^f = \bar{x}_0 + \delta_{i,0}, \tag{4.15}$$

where $x^f_{i,0}$ is the ith ensemble member at time $t = 0$, $\bar{x}_0$ is the mean initial state estimate, and $\delta_{i,0}$ represents the perturbation added to the mean state to generate the ith ensemble member.

Forecast Step

We have an observation set:

$$y_{i,k} = y_k + u_i, \tag{4.16}$$

with $u_i \sim \mathcal{N}(0, R)$, where R is the observation error covariance matrix. $y_{i,k}$ is the observable variable, number of infections, for the ith ensemble at time k. Let's assume we have m ensembles, and at each iteration k, we compute the ensemble means, where H is the observation model,

$$\begin{aligned} \bar{x}^f_k &= \tfrac{1}{m} \textstyle\sum_{i=1}^m x^f_{i,k} \\ \bar{u} &= \tfrac{1}{m} \textstyle\sum_{i=1}^m u_i \\ \bar{y}^f_k &= \tfrac{1}{m} \textstyle\sum_{i=1}^m H(x^f_{i,k}) \end{aligned} \tag{4.17}$$

and the normalized anomalies

$$\begin{aligned} [X_f]_{i,k} &= \tfrac{x^f_{i,k} - \bar{x}^f}{\sqrt{m-1}} \\ [Y_f]_{i,k} &= \tfrac{H(x^f_{i,k}) - u_i - \bar{y}^f_k + \bar{u}}{\sqrt{m-1}} \end{aligned} \tag{4.18}$$

From this, we can now compute the Kalman gain matrix:

$$K_k = X^f_k (Y^f_k)^T \{Y^f_k (Y^f_k)^T\}^{-1} \tag{4.19}$$

where X^f_k is the forecasted state estimate of the system at time k and Y^f_k is the predicted measurement at time k.

Update Step

For $i = 1, \ldots, m$

$$\begin{aligned} x_{i,k}^a &= x_{i,k}^f + K_k(y_{i,k} - H(x_{i,k}^f)) \\ x_{i,k+1}^f &= M(x_{i,k}^a) \end{aligned} \tag{4.20}$$

This is the basic idea behind the operation of EnKF, now we will go into the testing.

4.2 Results

As we have gone over the procedure behind the Kalman Filter and Ensemble Kalman Filter methods, we now utilize the built-in function EsembleKalmanFIlter from the filterpy.kalman library in Python. More information about the library and the function can be found in the Jupiter notebook published at [github].

Before diving into any specific model, we need to set up the general program that will be used across the three cases. We have the initial conditions for the total number of population, $N = 10000$, while $S(t) = 9999$, $I(t) = 1$ at time $t = 0$. We also adjust our prediction model daily in the first hundred days

$$t_{max} = 100 \text{ and } dt = 1 \tag{4.21}$$

The number of ensembles refers to the number of individual forecasts or simulations that are run in parallel, each representing a possible state of the system being studied. Then, these ensembles will be used to study the approximate probability distribution of the system's state at a given time. While a high number of ensembles can capture the full range of possible system states or uncertainty, this increases computational demands. Therefore, it is essential to choose an optimal number of ensembles, and in our case, we fix the number of ensembles to be 40 for the endemic and epidemic SIR models and will increase this number when dealing with a more complex system like

SIRW.

The setup for the initial covariance matrix, R, and process noise matrix, Q, is tricky as it depends on prior knowledge we have for our case. Moreover, our case, for example, the epidemic SIR model, will be running our program for the system in (2.1) plus

$$\begin{aligned} d_t\beta &= 0 \\ d_t\gamma &= 0 \end{aligned} \tag{4.22}$$

$$cov_{compartment} = 100, \quad cov_{paramter} = 0.01 \tag{4.23}$$

The high covariance values for the S, I, and R reflect a high degree of initial uncertainty in thee counts of susceptible, infectious, and recovered individuals due to under-reporting, asymptomatic cases, and delays in data collection. We set the initial covariance of β and γ to be low because we are confident with our initial guess for the parameters due to the previous estimates we have conducted above. However, these values can always be adjusted from case to case, and we will not go deeper into this topic in this paper. More information about the sensibility of the initial covariance matrix can be found elsewhere in other research.

$$Q = \begin{pmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0.001 \end{pmatrix} \tag{4.24}$$

0.01 represents the variance accounting for the random fluctuations in the number of each compartment, and 0.001 is the value of the variance for that of the parameters. For the sake of simplicity, we will keep these two values constant throughout our paper.

4.2.1 Endemic SIR Model

Without Noise

We generate our dataset with the endemic SIR system in (2.1) with $\beta = 0.3$ and $\gamma = 0.1$. Our initial guesses of β and γ are 0.1 and 0.5 respectively. Lastly, we will proceed to run our approach

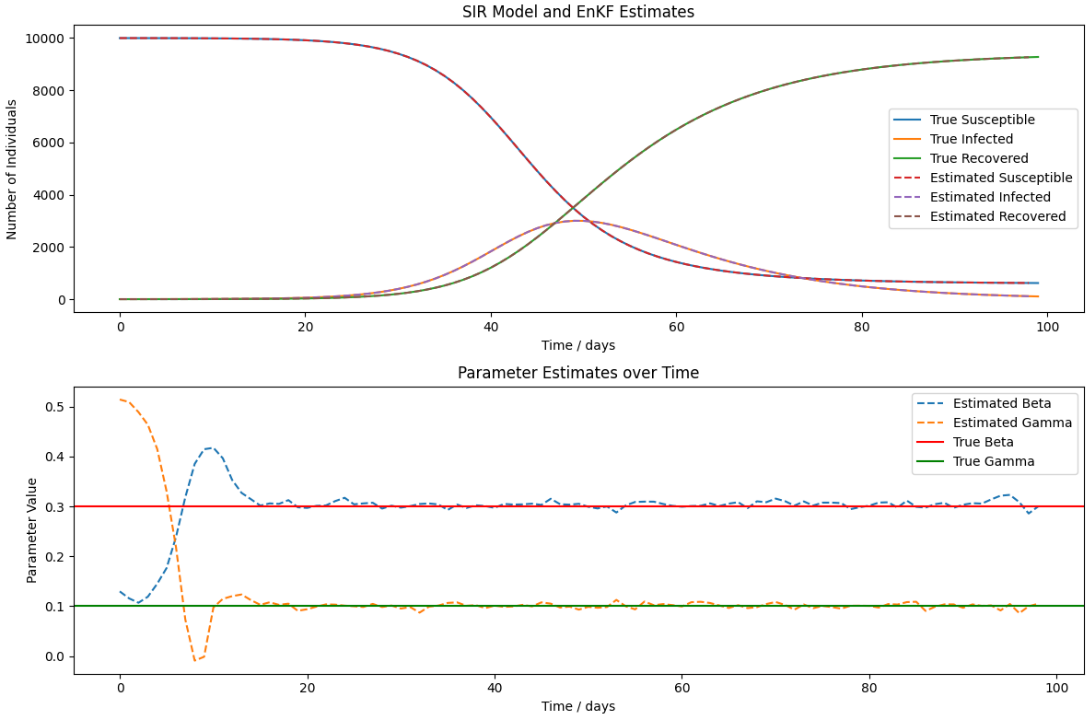

Figure 4.1: Simulation result of the endemic SIR model with the Ensemble Kalman Filter method

The second plot shows how the parameters are tuned, and in less than twenty days, the two parameters start to converge to the true values. In the first plot, our model forecasted the S, I, and R compartments perfectly, but this is not always a good sign, and we will see later why this is the case.

With Noise

The noise is introduced to our data:

$$z_S \sim \mathcal{N}(S_{\text{true}}[i], 10), \tag{4.25}$$

$$z_I \sim \mathcal{N}(I_{\text{true}}[i], 10), \tag{4.26}$$

$$z_R \sim \mathcal{N}(R_{\text{true}}[i], 10), \tag{4.27}$$

and our optimization gives the following result

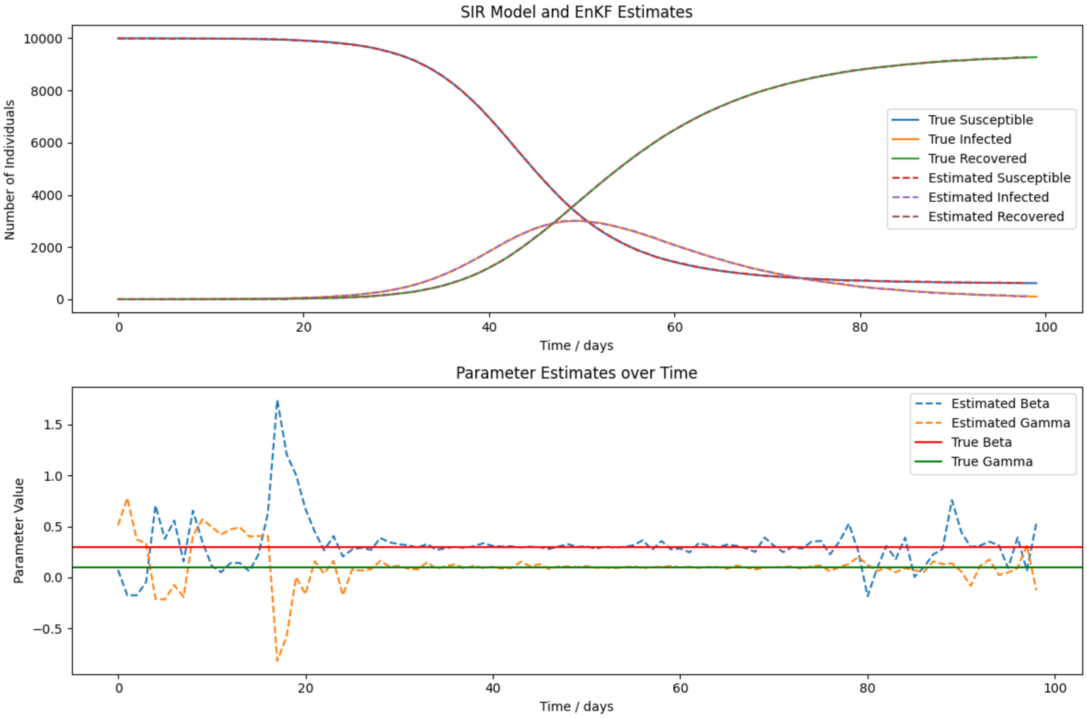

Figure 4.2: Simulation result of the endemic SIR model with the Ensemble Kalman Filter method with noise

Our approximation for parameters still holds regardless of some fluctuations within our first hundred days.

4.2.2 Epidemic SIR Model

For the epidemic SIR Model, our dataset is generated by the epidemic SIR system in (2.2) with $\beta = 0.3$, $\gamma = 0.1$, and $\mu = 0.001$. Our initial guess for the three parameters are

$$\beta = 0.2,\ \gamma = 0.01,\ \mu = 0.005 \tag{4.28}$$

Due to the sensibility of β, we also change our covariance for β in the initial covariance matrix to 9.

Without Noise

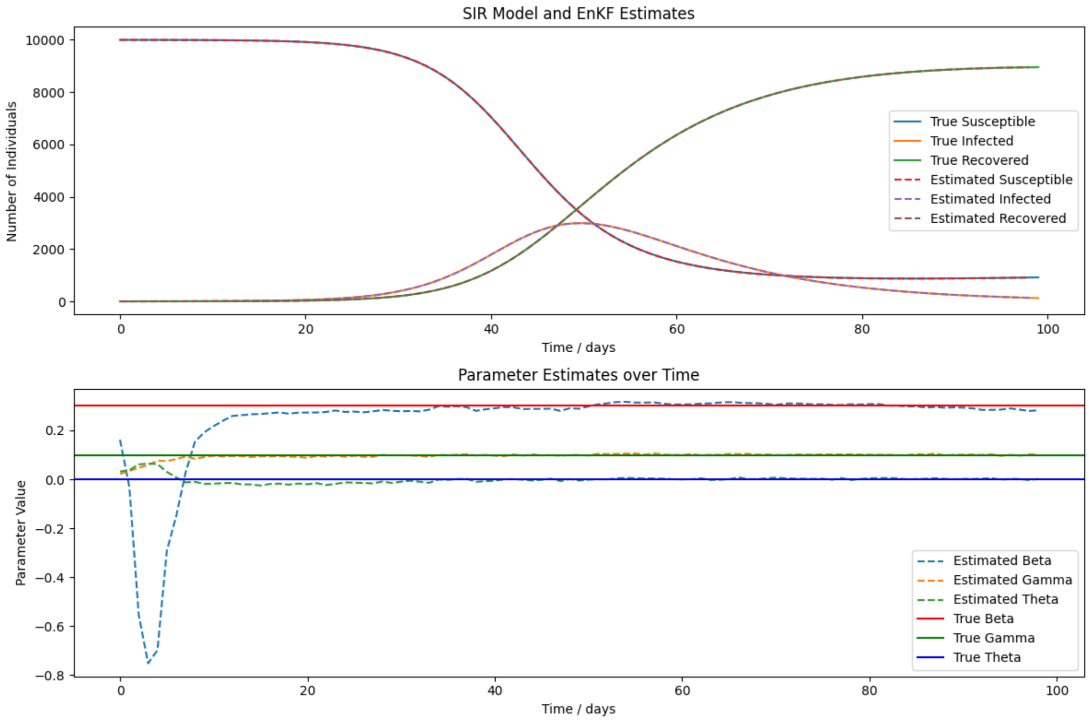

Figure 4.3: Simulation result of the epidemic SIR model with the Ensemble Kalman Filter method

In the second plot, the estimated value for β varied a lot at first then converged to its true value. As the number of optimized targets increases, our approach begins to give out more fluctuating results. This will be shown more clearly in the case of the

SIRW model.

With Noise

We now add noise into our data following:

$$z_S \sim \mathcal{N}(S_{\text{true}}[i], 1), \tag{4.29}$$

$$z_I \sim \mathcal{N}(I_{\text{true}}[i], 1), \tag{4.30}$$

$$z_R \sim \mathcal{N}(R_{\text{true}}[i], 1). \tag{4.31}$$

With such noise, our result turns out to be

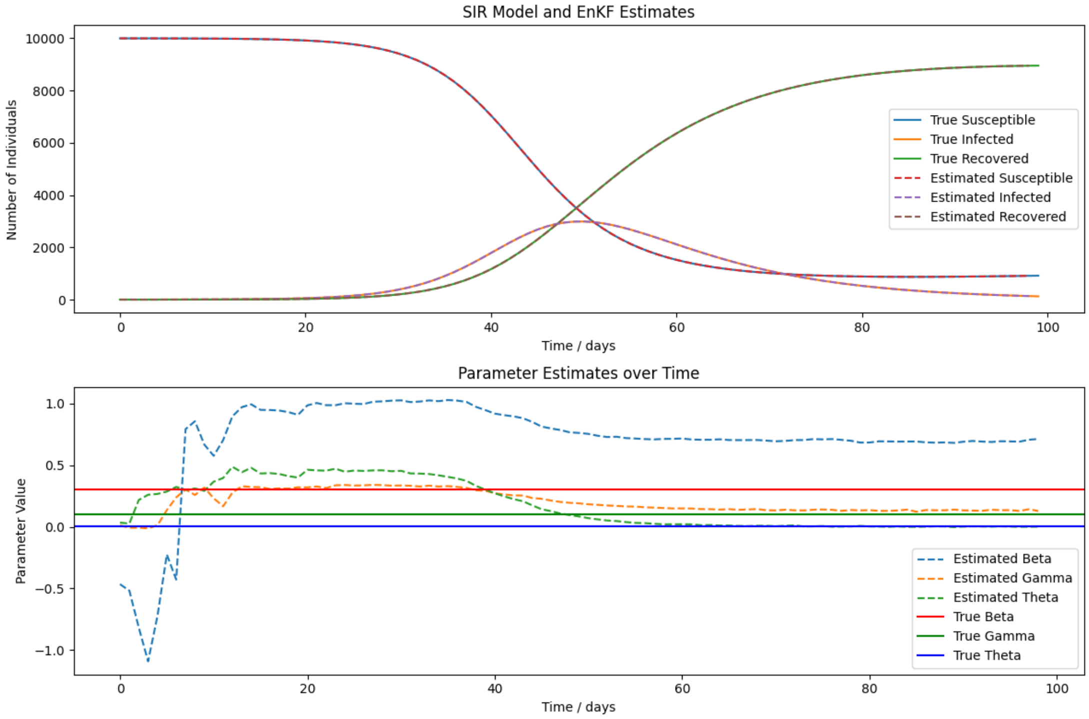

Figure 4.4: Simulation result of the epidemic SIR model with the Ensemble Kalman Filter method with noise

Although the introduction of even small white noise leads to an inaccurate approximation of our parameters, this can be controlled by adjusting the Q matrix. However, this is beyond the scope of this paper, and hopefully, we can address this issue in another paper.

4.2.3 SIRW Model

The SIRW model is the most complicated system in our paper, but surprisingly, although we keep everything about the initial covariance and Q matrix identical to what we mentioned in (4.23) and (4.24), the outcome is promising. The only thing we changed in our testing for the SIRW model that differs from that of the endemic SIR model is that we increased the number of ensembles from 40 to 400. This action is necessary, given the complexity of the SIRW model.

As before, we generate our dataset with the SIRW model in (2.3) with

$$\beta = 0.3,\ \gamma = 0.05,\ \theta = 0.1,\ \omega = 0.01,\ \alpha = 0.02. \tag{4.32}$$

Our initial guess is

$$\beta = 0.2,\ \gamma = 0.03,\ \theta = 0.12,\ \omega = 0.02,\ \alpha = 0.01. \tag{4.33}$$

Without Noise

Finally, we run our program:

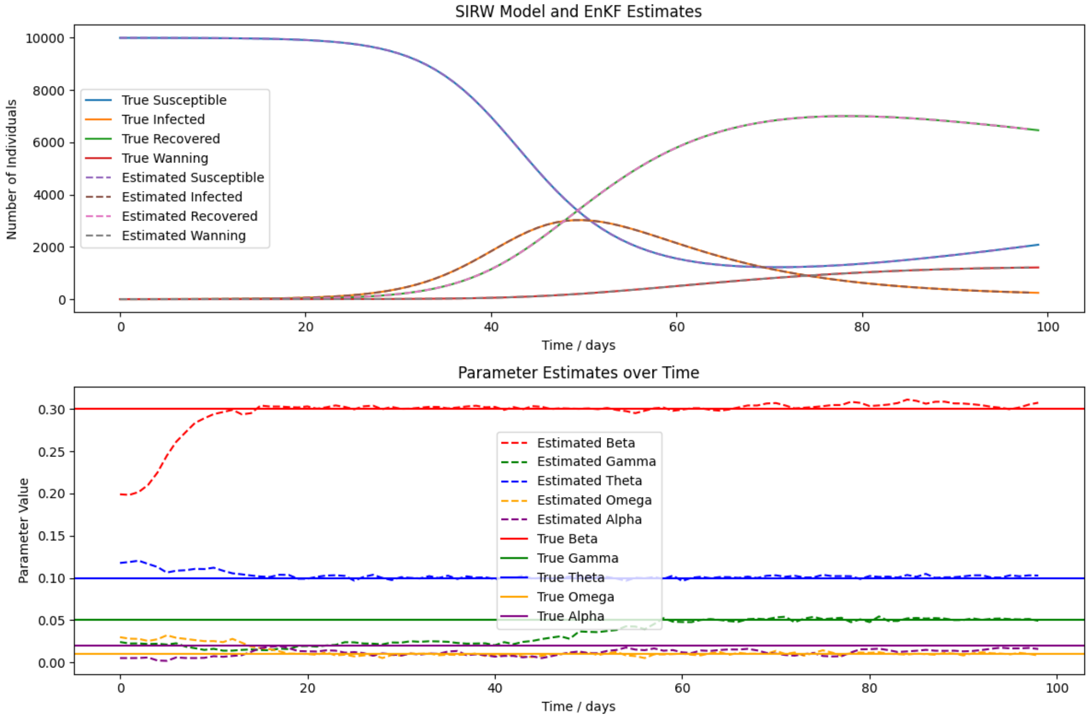

Figure 4.5: Simulation result of the epidemic SIRW model with the Ensemble Kalman Filter method

After the sixtieth day, all of our parameters converged to the true values presented in our generated dataset. This outcome gives hope to what the Ensemble Kalman Filter method can do when dealing with sophisticated, nonlinear models.

With Noise

Next step, we add some noise to our data:

$$z_S \sim \mathcal{N}(S_{\text{true}}[i], 1), \tag{4.34}$$

$$z_I \sim \mathcal{N}(I_{\text{true}}[i], 1), \tag{4.35}$$

$$z_R \sim \mathcal{N}(R_{\text{true}}[i], 1), \tag{4.36}$$

$$z_W \sim \mathcal{N}(W_{\text{true}}[i], 1). \tag{4.37}$$

We first introduce a little bit of noise to see how the optimization process deals with this

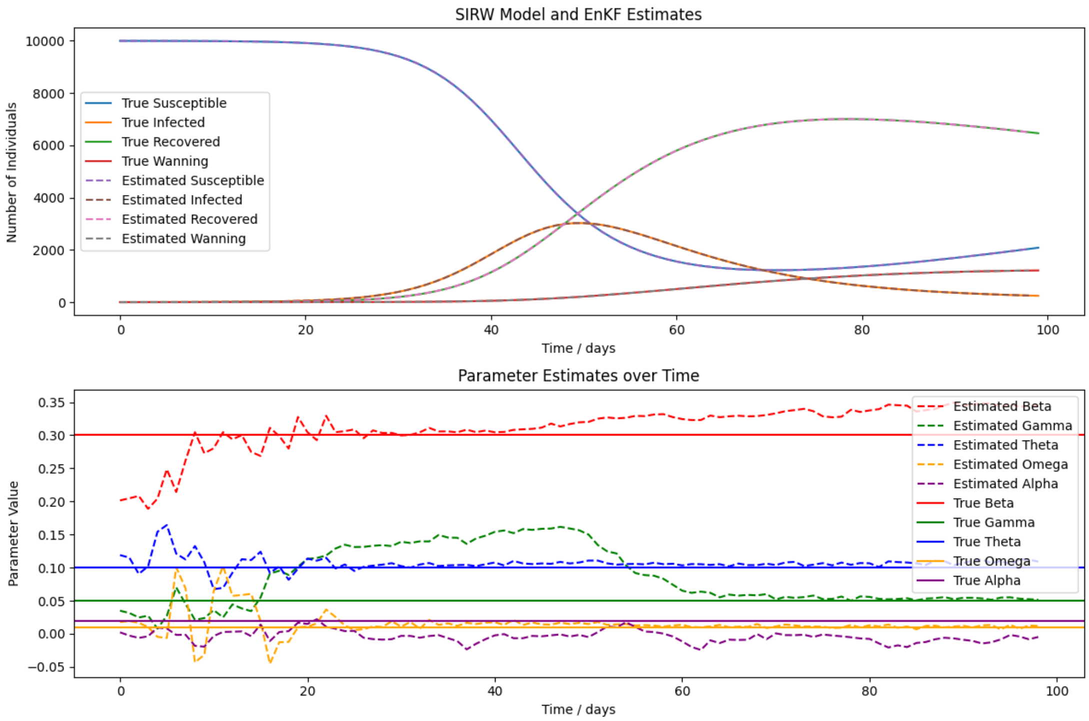

Figure 4.6: Simulation result of the epidemic SIRW model with the Ensemble Kalman Filter method with noise

Our parameter tuning seems fine with a small amount of white noise, but what if the data is noisier?

$$z_S \sim \mathcal{N}(S_{\text{true}}[i], 10), \tag{4.38}$$

$$z_I \sim \mathcal{N}(I_{\text{true}}[i], 10), \tag{4.39}$$

$$z_R \sim \mathcal{N}(R_{\text{true}}[i], 10), \tag{4.40}$$

$$z_W \sim \mathcal{N}(W_{\text{true}}[i], 10). \tag{4.41}$$

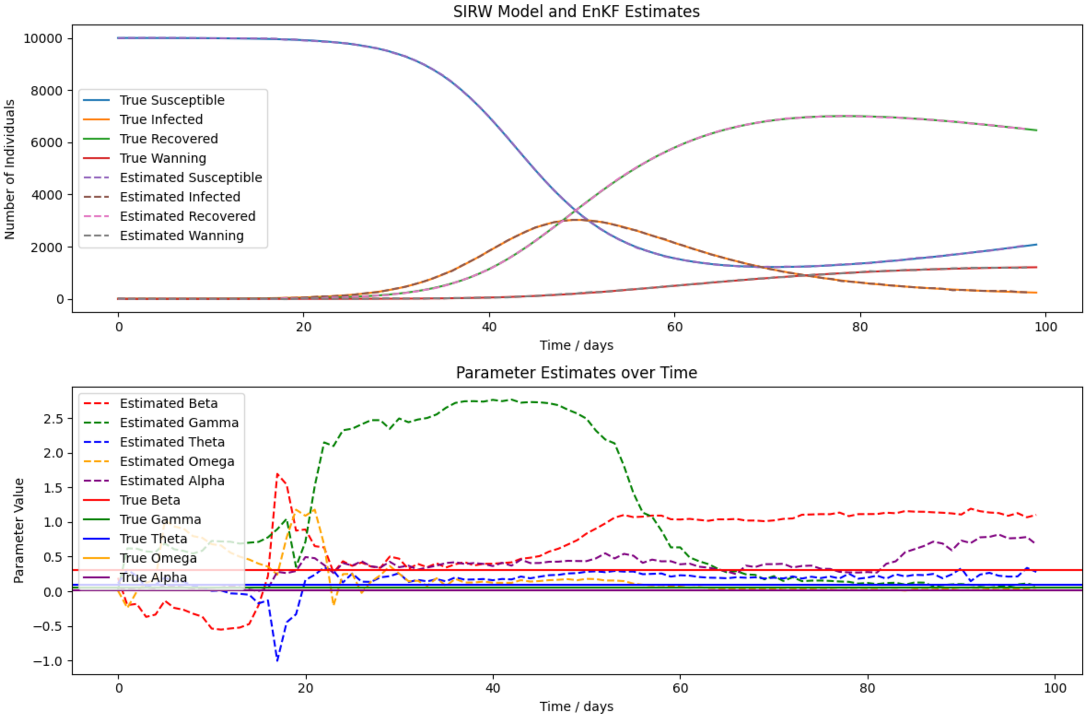

Figure 4.7: Simulation result of the epidemic SIRW model with the Ensemble Kalman Filter method with more noise

As the uncertainty increased, our parameter optimization did not perform well. However, keep in mind that the matrix Q that we have kept constant throughout our paper can potentially help with this situation.

Chapter 5

Conclusion

5.1 Discussion

5.1.1 Trust Region Method

Our deterministic approach with PyBOBYQA proves its practicality for basic models like epidemic and endemic SIR. This approach also showcases the capability to deal with noisy data. However, our derivative-free method failed to solve the SIRW model. The reason for this can be due to the complexity of the model. Another interesting result from this chapter 3 is about the multiple model case. The epidemic SIR model is less sophisticated than the endemic SIR model, and the endemic one was able to capture the behavior of the data generated by the epidemic SIR accurately. In other cases, the epidemic SIR model also performed well when attempting to model data that has an unknown variable to it — the role of parameter μ. Although this is an exciting finding, its practicality is still debatable. We have to answer the question of how accurate the estimated model by the epidemic SIR is and if it has any predicting power, how its precision will change over time.

5.1.2 Ensemble Kalman Filter

Our probabilistic approach with the Ensemble Kalman Filter is useful for the cases of the epidemic SIR model and the SIRW model but not the endemic SIR. Unlike our deterministic approach, the complexity of a model does not seem to have a significant impact on EnKF. Although producing excellent results for all three cases under the noise-free condition, our EnKF approach is sensitive to noise. Introducing a small amount of white noise to the endemic SIR, we saw its incapability to optimize β. However, the noise test with the SIRW model turned out to be quite precise for the case of adding little noise. Another noticeable result is that EnKF took longer to have the parameters converge to the actual values as the complexity of the model rises.

5.2 Future Perspective

We will investigate other features in the EnKF as the initial covariance or noise process matrix. We believe that these features affect the approach's sensitivity to noise. For example, if we have knowledge of how random our datasets are, we can alert our approach so that the updating will be less influenced by new information since it can be likely caused by noise. Moreover, we want to understand the reason behind the difference in performance under noisy datasets between the case of epidemic SIR and SIRW. It is not very intuitive to see the model performing better with more complex systems. Lastly, we want to see how the change in initial conditions will impact our prediction as a bad guess may result in a completely wrong or time consuming solution.

Bibliography

[1] M. Asch, M. Bocquet, and M. Nodet. *Data assimilation: Methods, algorithms, and applications*. SIAM, 2016.

[2] T. Bodnár, G. P. Galdi, and S. Nečasová. *Fluid-structure interaction and biomedical applications*. Birkhäuser, 2014.

[3] C. Cartis, J. Fiala, B. Marteau, and L. Roberts. Improving the flexibility and robustness of model-based derivative-free optimization solvers. *ACM Transactions on Mathematical Software*, 45(3):1–41, 2019.

[4] C. Cartis, L. Roberts, and O. Sheridan-Methven. Escaping local minima with local derivative-free methods: A numerical investigation. *Optimization*, 71(8):2343–2373, 2021.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. MPS-SIAM Series on Optimization. MPS/SIAM, Philadelphia, 2000.

[6] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8 of *MPS-SIAM Series on Optimization*. MPS/SIAM, Philadelphia, 2009.

[7] MJD Powell. The bobyqa algorithm for bound constrained optimization without derivatives. DAMTP 2009/NA06, University of Cambridge, Cambridge, UK, 2009.