Assignment 2 – Report

Implementation

First, a loss function was defined given an input of parameters-

The SEIR model was implemented with these parameters, similar to the first assignment using numerical difference method.

If s[t], e[t], i[t], r[t] respectively denote the number of people in the susceptible, exposed, infected and recovered state, then we have-

s[t+1] - s[t] = - $\beta i[t]s[t]$/N

e[t+1] - e[t] = $\beta i[t]s[t]$/N - αe[t]

i[t+1] - i[t] = αe[t] – γi[t]

r[t+1] - r[t] = γi[t]

Where N is the population and is equal to 70000000, α = $\frac{1}{5.8}$ and γ = $\frac{1}{5}$

For the first and third wave p vector was taken as [β,e[0],i[0],CIR] and r[0] was set to 0. In the second wave, p vector was [β,e[0],i[0],r[0],CIR]. s[0] was found as follows:

s[0]=N-e[0]-i[0]-r[0]

Thus the evolution of s,e,i,r values were found

The daily reported cases were found as αe[t]/CIR and the running average over 7 days were found.

Note: the simulation was started 6 days before the given time periods, and the cases on the first day as taken as an average over the past 7 days. Similarly, the running average was taken for the given case data for the corresponding time period

If data_avg denotes the average cases from data and model_avg denotes the average cases predicted by the model, then the loss function was defined as follows

$$l(P) = \frac{\sum_t(\log(average\ cases\ from\ data) - \log(cases\ predicted\ by\ model))^2}{Number\ of\ Days}$$

Now, this loss function l(P) was minimized by gradient descent.

The gradients were computed with the given perturbations in case of each parameter:

For example, gradient with respect to β was estimated by perturbing β by 0.01 while keeping the rest of the parameters constant

$$Grad(\beta) = \frac{lf(\beta + 0.01) - lf(\beta - 0.01)}{0.02}$$

Similarly, the perturbations for e[0],i[0] and r[0] were taken as 100 and CIR was perturbed by 0.1

Each of these gradients were also multiplied by an appropriate scaling factor in order to ensure that the gradients with respect to each parameter are of similar order. These factors were largely obtained by trial and error.

Lastly, the obtained gradient vector was normalized.

Now, the vector p was updated by the following equation:

$p' = p - \frac{grad(p)}{i+1}$, where i denoted the number of iterations.

Using if conditions, it was ensured that the parameters remained in physically realistic bounds and the ranges specified in the assignment.

Using a while loop, 20,000 iterations of gradient descent were run.

Since the outcome of the gradient descent seemed to be dependent on the initial guesses, first the algorithm was run with random initial values to obtain a rough range of the parameters. These were then inputted as the starting point of the gradient descent, in order to converge to the final answer and obtain a very low loss function.

For each wave, the results were printed in a text file and plots were made of the daily cases and model output, and evolution of s,e,i and r values

Results

First Wave

Through the gradient descent algorithm, a loss of 0.04 was obtained and the following parameters were found:

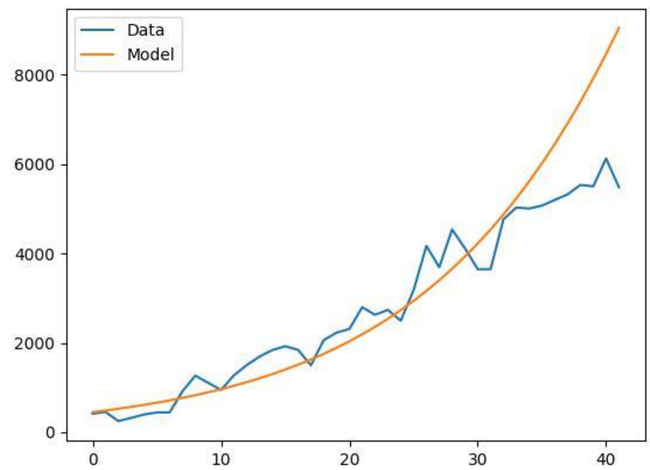
Beta: 0.407

E(0): 24787

I(0): 48117

CIR: 30

Plot of Daily Cases and Model Output



We observe that the model mostly predicts the case data quite well except towards the end where the cases predicted by the model shoot up. Some of the possible reasons for this may be

- We are modelling a constant contact rate beta. However, we know that I reality, beta was variable and tended to decrease after implementations of lockdowns and social distancing. This can explain why the daily case data seems to plateau while the model shoots up.
- Limitations in the implementation of gradient descent algorithm, as we have not been able to achieve the target loss of 0.01(discussed in more detail at the end)

Second Wave

Through the gradient descent algorithm, a loss of 0.0095 (less than 0.01) was obtained and the following parameters were found:

beta: 0.608

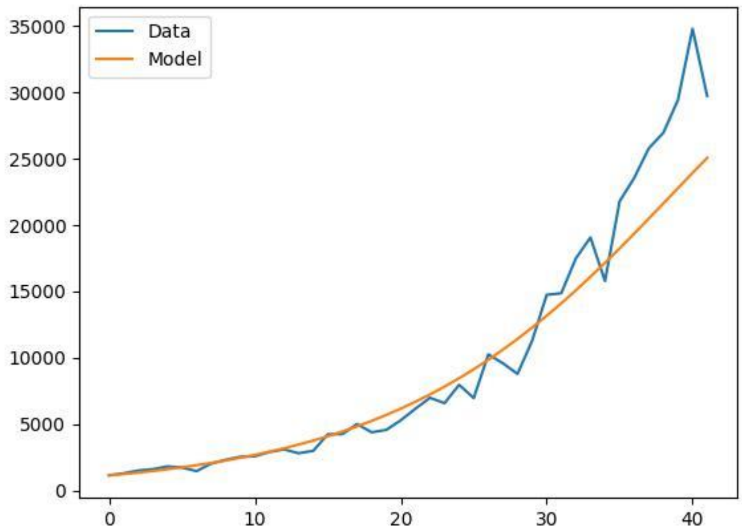
E(0): 79992

I(0): 89984

R(0): 18059984

CIR: 30

Plot of Daily Cases and Model Output



Once again, we observe that the model mostly predicts the case data quite well except towards the end. Similar to the first wave, one of the reasons for this disparity may be a variable beta, which in this case appears to be increasing. Since we have reached the target loss of below 0.01, one may expect a better fit of the model. However, it might also be possible that the particular expression we are using as the loss function is not perfect in characterizing how well our model fits the data, and we might obtain better fits with a different function.

Third Wave

Through the gradient descent algorithm, a loss of 0.03 was obtained and the following parameters were found:

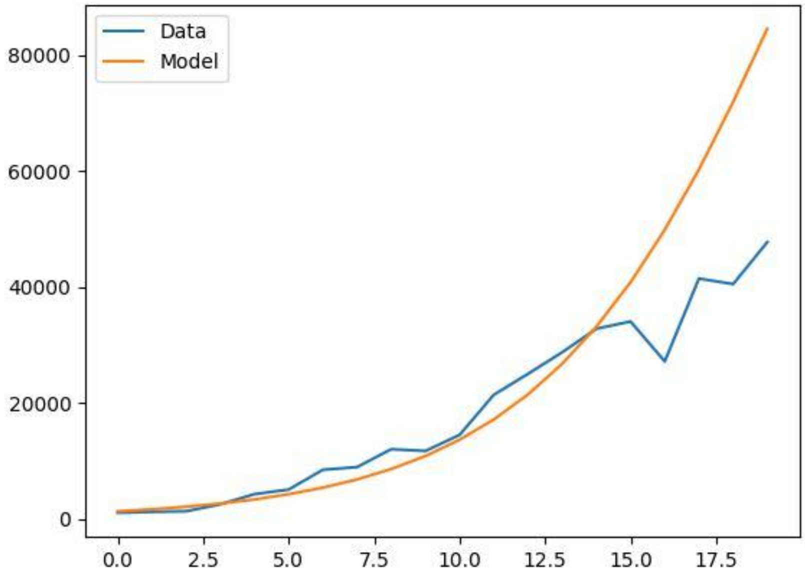
beta: 1.219

E(0): 12499

I(0): 34696

CIR: 30

Plot of Daily Cases and Model Output



Similar to the first wave, we can see that the model is able to predict the case data quite well except towards the end where the cases rise above the real data. The possible reasons for this are similar to those listed under the first wave.

Discussion on the Gradient Descent Algorithm

In case of both the first and third wave, the algorithm was unable to minimize the loss function to the target of 0.01, and the lowest loss obtained was 0.04 and 0.03 respectively. This could be due to a multitude of reasons. Firstly, since the loss function was exponential in beta, the resulting gradient for the beta parameter was very high compared to the others. This caused a need to manually scale the gradient values for all the parameters in order for the gradient descent to work as expected. Also, since the gradient descent algorithm detects local minima, the outcome was heavily dependent on the initial guesses for the parameters. Thus, the initial inputs and the scaling which had to be obtained through trial and error posed limitations on reducing the loss to 0.01.

Another thing that was observed was the CIR in all 3 cases was 30. This was likely because the CIR tended to shoot out of bounds and hence was automatically brought back to the boundary value. This is likely because of issues in the scaling of the gradients. However, the scaling values I converged to were the best I could obtain manually, and the gradient descent algorithm did not work properly when the gradient with respect to CIR was scaled down. One way to combat these issues is come up with a rigorous method of scaling the gradients which also varies with time and targets the optimization of the necessary parameters. It can be noted that in spite of this issue, we were able to get quite low losses. It was observed that the CIR did not affect the value of the loss function as much as the other parameters.