

Data Analytics Assignment - 5

G Ramesh Babu
SR No. 20950

Covid-19 Modeling

Abstract

This Report shows the analysis done on Covid-19 Modeling as part of Data Analytics Assignment - 5.

1 Implementation Summary

1.1 Importing Data

The function `import_data()` imports COVID-19 data from the specified directory and does the following.

- Removes unwanted columns ('Other', 'Total Doses Administered', 'Recovered', 'Second Dose Administered', 'Deceased').
- Transforms the date column to a datetime format.
- Selects two specific date ranges for analysis (from March 8, 2021, to April 26, 2021) and (from March 8, 2021, to Sept 19, 2021). One week extra data is taken to compute the differences and running averages
- Calculates the differences in confirmed cases and the number of tests conducted, converting them to integer values.
- Ensures that the data is properly indexed.
- Returns the preprocessed data as two NumPy arrays: `ModelData` and `OriginalData`.

1.2 Finding Running Averages

The function `RunningAverages(ModelData, OriginalData)` performs the following tasks:

- Calculates the running average for confirmed cases (`ConRunAvg`) by applying a convolution with a 7-day uniform filter.
- Computes the running average for the number of tests conducted (`Tested`) by taking the difference between consecutive values and dividing by 7 for each row.
- Extrapolates the data for test counts to predict values until December 31, 2021, and appends the results to the original 'tested' data.
- Calculates the running average for the number of first doses administered (`First`) by taking the difference between consecutive values and dividing by 7 for each row.
- Extrapolates the data for first doses administered to predict values until December 31, 2021, and appends the results to the original 'first' data.
- Computes the ground truth data (`GrouTruth`) by taking the difference between consecutive values and dividing by 7, resulting in the daily average of confirmed cases.

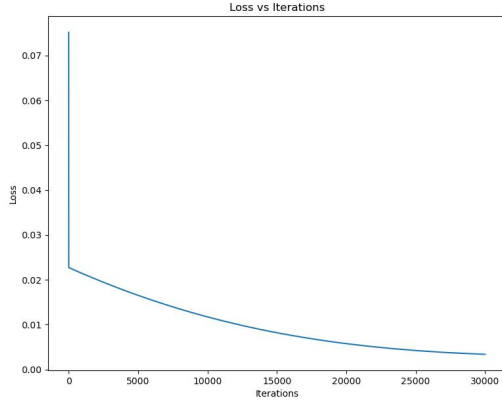
1.3 Implementing SEIRV Model

The ‘SEIRV’ class is a Python implementation of the SEIRV compartmental model for simulating the progression of diseases like COVID-19. It is equipped with various functions and attributes to facilitate modeling and analysis. Here are the key functionalities:

- **Initialization:** The class is initialized with several parameters that define the initial conditions and characteristics of the disease. These parameters include:
 - Initial populations of susceptible(S_0), exposed(E_0), infected(I_0), and recovered(R_0) individuals and CIR_0 and β
 - Total days for simulation.
 - Running averages of observed data.
 - Parameters α, γ, ϵ and Population N are also initialized
- **Time Series Generation:** The `generate_time_series()` method simulates the SEIRV model over a specified duration (days). For each day, it calculates the variation in the number of individuals moving from the R (recovered) compartment to the S (susceptible) compartment, considering immunity waning over time. This method also generates time series data for various population compartments, including S, E, I, and R. Additionally, it computes rolling averages to smoothen the data.
- **Loss Computation:** The `calculate_loss()` method evaluates the model’s performance by computing a loss function. To calculate this loss, it computes the mean squared error between the natural logarithm of Running Average of Confirmed cases from the given data and also from the Model. The result is normalized by dividing by the number of days.
- **Gradient Calculation:** Gradients are computed by perturbing the parameters and finding the change in the loss wrt to the change in the change in the respective parameter. This is done parameter wise.
- **Parameter Optimization:** `optimize_params()` optimizes model parameters, including beta (transmission rate), initial conditions, and CIR_0 , to minimize the loss function using the gradients computed before.
- **Future Predictions:** The class forecasts the future progression of the disease, considering closed-loop control and changes in beta. It can predict the fraction of susceptible people or new cases.
- **Plotting Results:** The `plotSEIR()` method creates visualizations of the SEIRV model’s results, including population compartments (S, E, I, R) and future predictions.

2 Results

2.1 Question 1-2



(a) Training the Parameters for SEIRV Model

Iteration	Loss
0	0.07518
5000	0.01654
10000	0.01174
15000	0.00821
20000	0.00578
25000	0.00424
30000	0.00340

(b) Loss vs Iterations

Model Parameters are initialized as follows:

- $\mathbf{N} = 7 \times 10^7$, $\beta_0 = 0.45$, $\mathbf{S}_0 = 0.7 \times N$, $\mathbf{E}_0 = 0.001 \times N$, $\mathbf{I}_0 = 0.001 \times N$, $\mathbf{R}_0 = 0.298 \times N$, $\mathbf{CIR}_0 = 15$

After training for 30000 iterations, the Model Parameters are updated as follows:

- $\beta_0 = 0.453752037906148$
- $S_0 = 49000000.0$
- $E_0 = 70000.0$
- $I_0 = 70000.0$
- $R_0 = 20860000.0$
- $CIR_0 = 12.04$

The Final Loss Obtained with these parameters is **0.0034** and these parameters are fixed for future prediction

2.2 Question 3

The following plots show the predictions obtained by open loop and closed loop controls for different values of β . The Actual reported cases are also shown till 20th Sep 2021

2.2.1 27th April 2021 - 20th Sep 2021

- Higher β values lead to increased infections, but the rapid growth of the exposed, infected, and recovered populations causes a swift decline due to the quick reduction in susceptible individuals.
- In $1/2 * \beta$ and $1/3 * \beta$ cases, daily infections sharply decrease and approach zero. This suggests that lower contact rates hinder rapid disease spread, ultimately leading to the pandemic's decline.
- The Actual reported cases consistently lags behind predictions, as reported cases tend to significantly underestimate actual cases, especially during the pandemic's peak.
- The Closed-Loop control scenario exhibits distinctive ruggedness with sharp fluctuations, driven by dynamic β adjustments based on the prevailing conditions.

- we can see that the susceptible population decreases more rapidly with higher β . This is because a higher β value results in a rapid infection of a larger portion of the population, leading to a quicker reduction in the susceptible population (those not yet exposed to the virus).

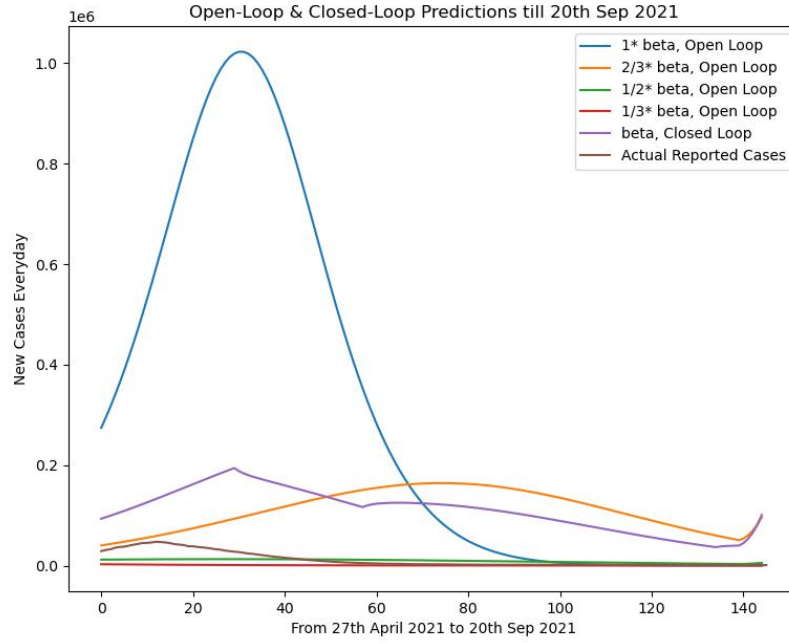


Figure 2: Number of New Cases Predicted Every Day

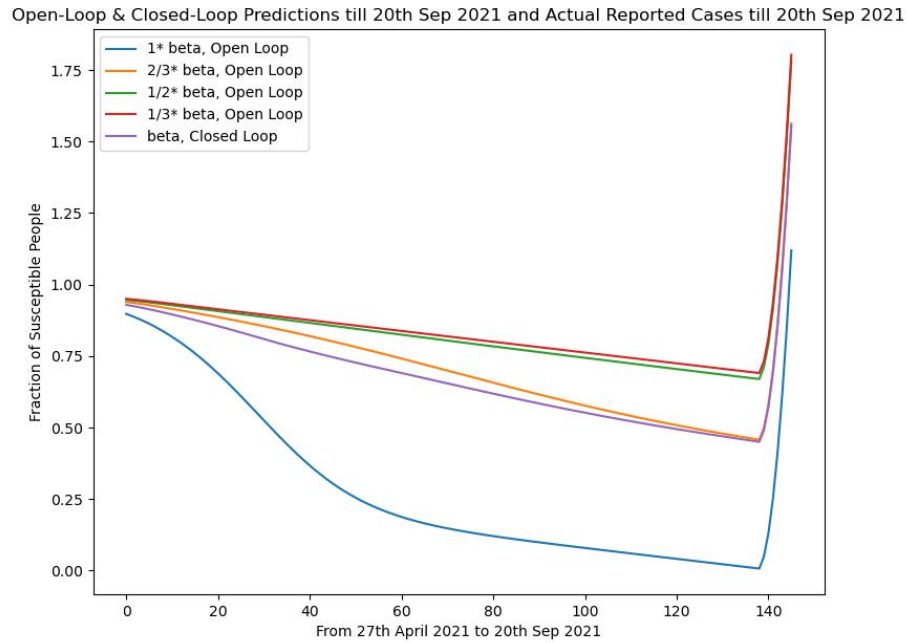


Figure 3: Evolution of the Fraction of Susceptible People

2.2.2 27th April 2021 - 31st Dec 2021

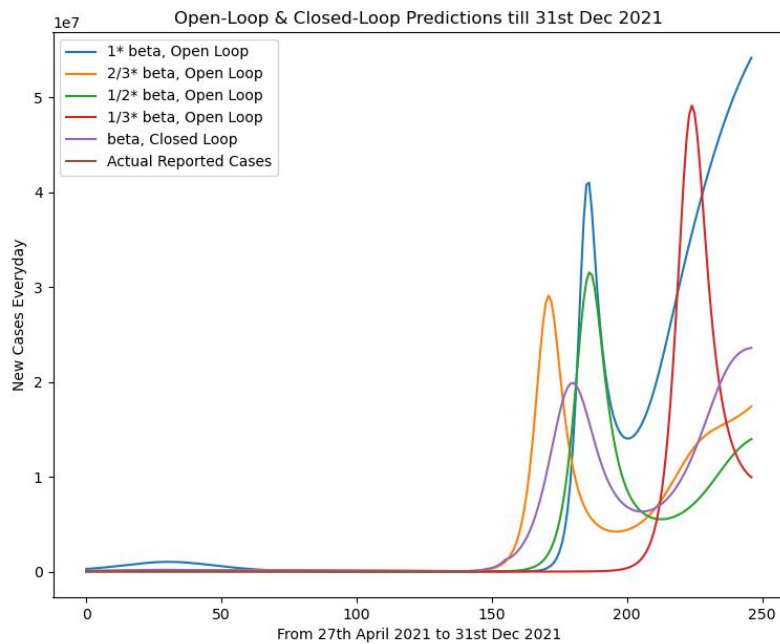


Figure 4: Number of New Cases Predicted Every Day

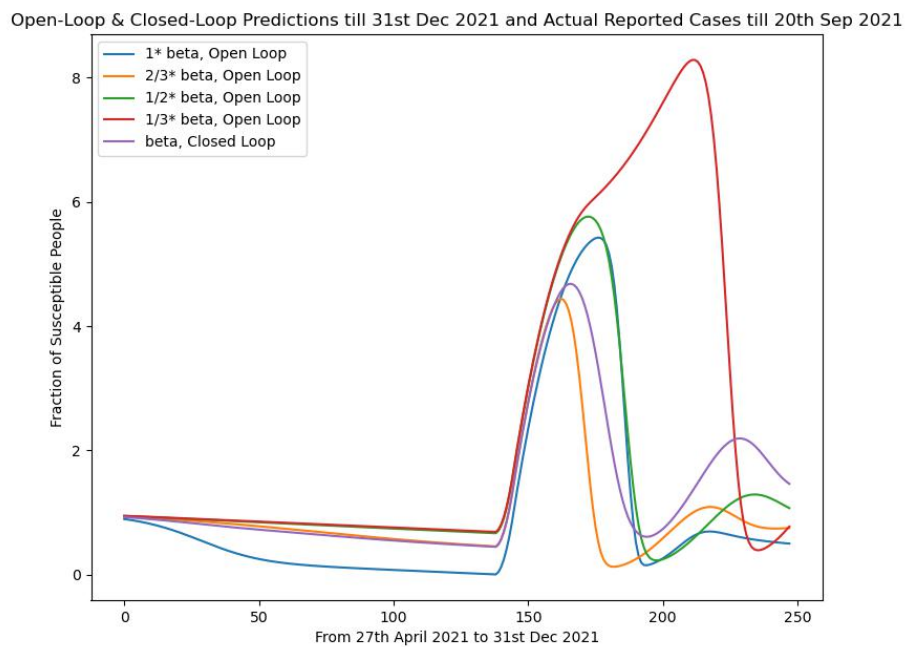


Figure 5: Evolution of the Fraction of Susceptible People