

# Spatial-temporal Prediction of COVID-19 Case Counts Through Epidemiology Model

Wang, Caiwei

Wang, Shuyuan

March 8, 2021

## 1 Introduction

### 1.1 Problem

The early state and late stage of a pandemic is very different. At early stage, the case number grows exponentially. Government agencies and institutions want to the ability to forecast the number of cases in order to allocated medical and other resources. Furthermore, knowing how protocols such as stay at home order can affect the case number is extremely useful to make predictions. However, in order to predict number of cases in the future, the growth factor is needed and can be generated from fitting previous data to an epidemiology model. The exponential factor is based on two factors that can be learned from data: the infection rate,  $\beta$  and number of days a patient stays infectious  $D$ .

### 1.2 Context

One of the pressing problems in epidemiology is long term prediction of the spreading of an infectious disease. Of particular interest is how mitigation measures (government policies) can affect the number of infected in the future. Numerous efforts have been tried around the world. Many cities and states in the U.S. have ordered stay-at-home policies. It is useful to see how administering these orders can affect the case numbers, how would the case numbers react if the government revoke the orders.[1] I

## 2 Data

We use data from JHU's public COVID-19 GitHub repository [2], and mobility data provided by Descarted Lab [3].

### 2.1 Collection

The U.S. confirmed, death and recovered case is updated daily by regional health departments of different jurisdictions, then collected by JHU, Because all the data is U.S. (unlike comparing data among different countries), the testing method and data reporting method are relatively uniform and reliable to make inferences on.

The mobility data is allocated from many sources. For example, Apple Map can estimate the distance a typical person travel in that region from their back-end data of users moving in that geological areas.

### 2.2 Description

The pandemic hits different jurisdictions at a different time. Because the data set is collected by jurisdictions, we are able to calculate growth factor and make predictions at county or state level.

In order to fit data to the epidemiology model, we need to have time series data of three variables, estimated from confirmed, death, and recovered case numbers.

$$\begin{aligned} \text{Infected, } I &= \text{Confirmed Cases} \\ \text{Susceptible, } S &= \text{Population} - \text{Confirmed Cases} \\ \text{Removed, } R &= \text{Recovered} + \text{Deaths} \end{aligned}$$

The mobility data is representing the distance a typical member of a given population moves in a day. With this data set, we are able to see how stay-at-home orders by different states and the pandemic itself have effect on average mobility trends. The data set also provides additional information for our epidemiology model. [We haven't touched on this yet]

## 3 Methods

### 3.1 Overview

We decided to predict the case number in 3/2/2021 of all the counties in California. We first use gradient descent method to find infection rate and infection duration.

However, in reality, the case numbers of the entire country/state is not evenly distributed among counties. Besides computing the individual infection rate for all the counties, each as a separate entity from its neighbors, we will predict case numbers based on the mobility data provided by Descartes Labs[3] (how fast is people moving inside each county and across county boundaries), and the case numbers of each county's neighboring counties. We will use a  $dt$  around 0.001 day instead of 1 day to better predict the dynamics. This process requires a fixed  $\beta$  and  $D$  predetermined for each county.

Furthermore,  $\beta$  and  $D$  are also dynamic. So in the future, we will replace the fixed  $\beta$  and  $D$  with dynamic, changing according to the data.

### 3.2 Inferring Parameters

See Appendix 5.1 for code. We are using gradient descent to solve for  $\beta$  and  $\frac{1}{D}$ , infection rate and 1 over days staying infected. Given:

$$\begin{aligned} \xi &= \frac{1}{D} \\ f_s(I_n, N, S_n) &= -\beta \left( \frac{I_n}{N} \right) S_n, \\ f_I(I_n, N, S_n) &= -I\xi + \beta \left( \frac{I_n}{N} \right) S_n, \\ f_R(I_n) &= I_n\xi, \\ h &= 1 \end{aligned}$$

Loss function can be calculated by:

$$\frac{1}{N} \sum_{n=1}^N \nabla_{\theta} \left( \left( \frac{s(n+1)-s(n)}{h} - f_s(s(n), I(n), R(n); \theta) \right)^2 + \left( \frac{I(n+1)-I(n)}{h} - f_I(s(n), I(n), R(n); \theta) \right)^2 + \dots \right)$$

To calculate the above the term, we need to use the chain rule to differentiate with respect to  $\beta$  and  $\xi$

$$\nabla_{\beta} = 2 \cdot \left( \frac{S_{n+1}-S_n}{h} - (-\beta S_n \frac{I_n}{N}) \right) \cdot (S_n \cdot \frac{I_n}{N}) + 2 \cdot \left( \frac{I_{n+1}-I_n}{h} - (-\xi_k I_n + \beta_k \frac{I_n}{N} S_n) \right) \cdot (-S_n \cdot \frac{I_n}{N})$$

$$\nabla_{\xi} = 2 \cdot \left( \frac{I_{n+1}-I_n}{h} + (I_n \xi_k - \beta_k \frac{I_n}{N} S_n) \right) \cdot (I_n) + 2 \cdot \left( \frac{R_{n+1}-R_n}{h} - I_n \xi_k \right) \cdot (-I_n)$$

First initialize  $\theta$  at  $\beta = 0.2$  and  $\xi = 0.1$

Then, at each iteration update  $\beta$  and  $\xi$  according to the rules below

$$\beta_{k+1} = \beta_k - h_G \partial_\beta L(\theta | s(1), \dots, s(N)),$$

$$\xi_{k+1} = \xi_k - h_G \partial_\xi L(\theta | s(1), \dots, s(N)).$$

where the learning rate,  $h_G = 1/N$ . where  $N$  is the population  
When  $\beta$  and  $D$  both converge (a.k.a is the same as the previous iteration), we stop the iterations and return the two value, in order to fit into the ODE model.

### 3.3 Determining Learning Rate

Lipschitz continuous gradient condition is essential to ensuring convergence of many gradient decent based algorithms[4].The step size should scale inversely with the Lipschitz contant. We can calculate the constant by taking the eigenvalue of the hessian matrix of  $\theta$  See 5.1 to see code

The gradients are:

$$\nabla_\beta = 2 \cdot \left( \frac{S_{n+1} - S_n}{h} - \left( -\beta_k S_n \frac{I_n}{N} \right) \right) \cdot \left( S_n \cdot \frac{I_n}{N} \right) + 2 \cdot \left( \frac{I_{n+1} - I_n}{h} - \left( -\xi_k I_n + \beta_k \frac{I_n}{N} S_n \right) \right) \cdot \left( -S_n \cdot \frac{I_n}{N} \right)$$

$$\nabla_\xi = 2 \cdot \left( \frac{I_{n+1} - I_n}{h} + \left( I_n \xi_k - \beta_k \frac{I_n}{N} S_n \right) \right) \cdot (I_n) + 2 \cdot \left( \frac{R_{n+1} - R_n}{h} - I_n \xi_k \right) \cdot (-I_n)$$

The Hessian matrix will look like the following:

$$\begin{bmatrix} \frac{\partial \nabla_\beta}{\partial \beta} & \frac{\partial \nabla_\xi}{\partial \beta} \\ \frac{\partial \nabla_\beta}{\partial \xi} & \frac{\partial \nabla_\xi}{\partial \xi} \end{bmatrix}$$

We can calculate the hessian matrix given  $S_n, I_n$  and population  $N$ .  $n$  represents the timestamp, in our case, is the number of the day after the first day in our sequence.  $T$  stands for the total number of days in the sequence of data.

$$\frac{\partial \nabla_\beta}{\partial \beta} = \frac{1}{T} \sum_{n=1}^T \left( 2 \cdot \left( S_n \cdot \frac{I_n}{N} \right)^2 + 2 \cdot \left( S_n \cdot \frac{I_n}{N} \right)^2 \right)$$

$$= \frac{1}{T} \sum_{n=1}^T \left( 4 \cdot \left( S_n \cdot \frac{I_n}{N} \right)^2 \right)$$

$$\frac{\partial \nabla_\xi}{\partial \xi} = \frac{1}{T} \sum_{n=1}^T (2 \cdot I_n^2 + 2 \cdot I_n^2)$$

$$= \frac{1}{T} \sum_{n=1}^T 4 \cdot I_n^2$$

$$\frac{\partial \nabla_\xi}{\partial \beta} = \frac{1}{T} \sum_{n=1}^T -2 \cdot S_n \cdot \frac{I_n^2}{N}$$

$$= \frac{\partial \nabla_\beta}{\partial \xi}$$

### 3.4 Adding Mobility and Location Variable

After checking the accuracy of our model fitting process, we would calculate infection rates  $\beta$  for all the counties. Then we want to better predict the case numbers with more data: mobility and geographic information. Using Geographical Information And Mobility Data to Predict County Infection are necessary for improving the accuracy of our model. In order to find nearby counties, We used data provided by US Census to find out each counties' neighbors. In order to achieve a more accurate prediction, the direction of mobility is necessary. Therefore, we initially set out four closest counties in west-north-east-south direction.

for two neighboring points:  $x^1, x^2$ .

$$\Delta_x I(x, t)|_{(x^1, x^2, x^3)} \approx \frac{\partial_{x_1} I(x, t)|_{(x^1, x^2)} - \partial_{x_1} I(x, t)|_{(x^2, x^3)}}{x_1^1 - x_1^3} + \frac{\partial_{x_2} I(x, t)|_{(x^1, x^2)} - \partial_{x_2} I(x, t)|_{(x^2, x^3)}}{x_2^1 - x_2^3} \quad (1)$$

$$+ \frac{\partial_{x_1} I(x, t)|_{(x^4, x^2)} - \partial_{x_1} I(x, t)|_{(x^2, x^5)}}{x_1^4 - x_1^5} + \frac{\partial_{x_2} I(x, t)|_{(x^4, x^2)} - \partial_{x_2} I(x, t)|_{(x^2, x^5)}}{x_2^4 - x_2^5} \quad (2)$$

If  $x$  is a uniform mesh, then at  $x_{(2,2)}$ ,

$$\Delta_x I = \frac{(I(x_{(1,2)}, t) - I(x_{(2,2)}, t) - (I(x_{2,2}, t) - I(x_{(3,2)}, t)))}{\Delta x^2} + \frac{I(x_{(2,1)}, t) + I(x_{(2,3)}, t) - 2I(x_{2,2}, t)}{\Delta x^2} \quad (3)$$

$$= \frac{I(x_{(1,2)}, t) + I(x_{(3,2)}, t) - 2I(x_{2,2}, t)}{\Delta x^2} + \frac{I(x_{(2,1)}, t) + I(x_{(2,3)}, t) - 2I(x_{2,2}, t)}{\Delta x^2}. \quad (4)$$

$$\left. \frac{\partial I(x, t)}{\partial t} \right|_{t=t_2} \approx \frac{I(x, t_1) - I(x, t_2)}{t_1 - t_2}$$

$$\frac{I(x_2, y_2, t_1) - I(x_2, y_2, t_2)}{t_1 - t_2} = \frac{I(x_1, y_2, t_2) + I(x_3, y_2, t_2) - 2I(x_2, y_2, t_2)}{\Delta x^2} + \frac{I(x_2, y_1, t_2) + I(x_2, y_3, t_2) - 2I(x_2, y_2, t_2)}{\Delta y^2}.$$

$$\Delta x = \frac{x_3 - x_1}{2}, \quad \Delta y = \frac{y_3 - y_1}{2}.$$

Assume that  $t_1 - t_2 = 1$ , and that

$$I(x_2, y_2, t_1) = I(x_2, y_2, t_2) + I(x_1, y_2, t_2) + I(x_3, y_2, t_2) - 2I(x_2, y_2, t_2) + I(x_2, y_1, t_2) + I(x_2, y_3, t_2) - 2I(x_2, y_2, t_2).$$

What to do at  $(x_1, y_2)$ ?

$$I(x_1, y_2, t_1) = I(x_1, y_2, t_2) + I(x_0, y_2, t_2) + I(x_2, y_2, t_2) - 2I(x_1, y_2, t_2) + I(x_1, y_1, t_2) + I(x_1, y_3, t_2) - 2I(x_1, y_2, t_2).$$

### 3.5 Accuracy

After getting the  $\beta$  and  $D$ , we want to plug in these two values into an ODE model to check whether this model can predicts infection numbers of that specific region. In theory, the curve of the ODE model should fit our training data.

## 4 Results

We obtained the  $\beta$  and  $D$  from the 1/21/2021 to 3/1/2021 data from the JHU dataset. The prediction for 3/2/2021 and its comparison to the actual case count are as followed:

	County Name	Actual Case	Predicted Case	Percent Difference
0	Alameda	80873	80880	0.000090
1	Alpine	82	108	0.324086
2	Amador	3469	5780	0.666425
3	Butte	10981	11008	0.002475
4	Calaveras	1911	2041	0.068054
5	Colusa	2139	2157	0.008545
6	Contra Costa	62818	62983	0.002642
7	Del Norte	1007	1020	0.013619
8	El Dorado	9168	10500	0.145294
9	Fresno	95677	95793	0.001217
10	Glenn	2232	2358	0.056717
11	Humboldt	3219	3223	0.001315
12	Imperial	26913	3399386	125.310203
13	Inyo	1317	2522	0.915076
14	Kern	103622	104930	0.012623
15	Kings	22091	22215	0.005640
16	Lake	3164	3176	0.003803
17	Lassen	5623	5648	0.004619
18	Los Angeles	1194333	1194402	0.000058
19	Madera	15505	15601	0.006230
20	Marin	13261	13936	0.050909
21	Mariposa	395	527	0.334430
22	Mendocino	3821	3820	-0.000256
23	Merced	29195	29389	0.006650
24	Modoc	459	476	0.038033
25	Mono	1214	1227	0.011471
26	Monterey	42316	42443	0.003019
27	Napa	9056	9205	0.016508
28	Nevada	3979	9963	1.504130
29	Orange	261608	426252	0.629356
30	Placer	19882	20009	0.006434
31	Plumas	653	677	0.037004
32	Riverside	290325	238244	-0.179388
33	Sacramento	93678	93800	0.001305
34	San Benito	5772	6200	0.074179
35	San Bernardino	286814	287290	0.001660
36	San Diego	261001	261555	0.002125
37	San Francisco	34318	34438	0.003504
38	San Joaquin	67040	67091	0.000762
39	San Luis Obispo	19724	19821	0.004942
40	San Mateo	39096	39331	0.006034
41	Santa Barbara	32087	32206	0.003713
42	Santa Clara	110911	110948	0.000337
43	Santa Cruz	14700	14750	0.003444
44	Shasta	11045	-6331	-1.573222
45	Sierra	100	191	0.913028
46	Siskiyou	1779	1824	0.025715
47	Solano	30163	30382	0.007261
48	Sonoma	28222	28317	0.003368
49	Stanislaus	56323	51603	-0.083802
50	Sutter	8886	9031	0.016414
51	Tehama	5104	5119	0.003040
52	Trinity	372	53619	8.728851
53	Tulare	48086	48427	0.007104
54	Tuolumne	3963	3931	-0.007859
55	Ventura	77849	80428	0.033131
56	Yolo	12854	12935	0.006360
57	Yuba	5771	5837	0.011467

Actual cases on California, 3/2/2021

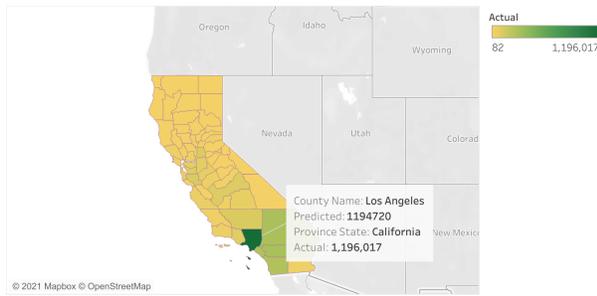


Figure 1: Actual Cases of California, 3/2/2021

Predicting cases on California, 3/2/2021

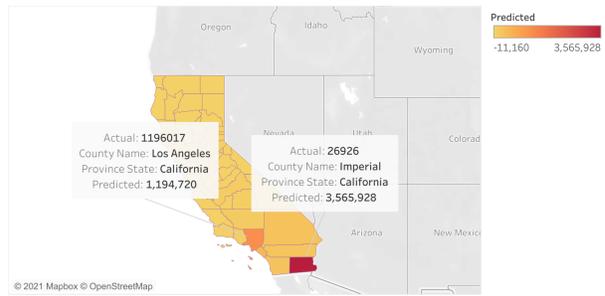


Figure 2: Predicting Cases of California, 3/2/2021

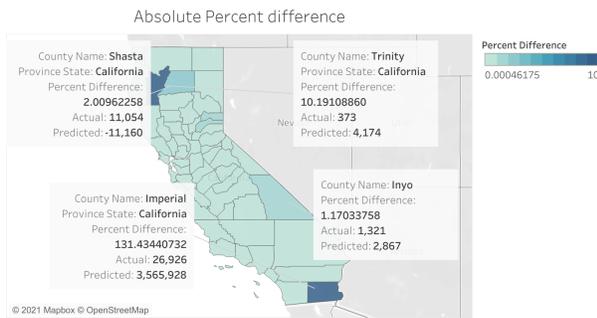


Figure 3: Heatmap of Absolute Percent Difference Between Actual Cases and Predicting Cases of California, 3/2/2021

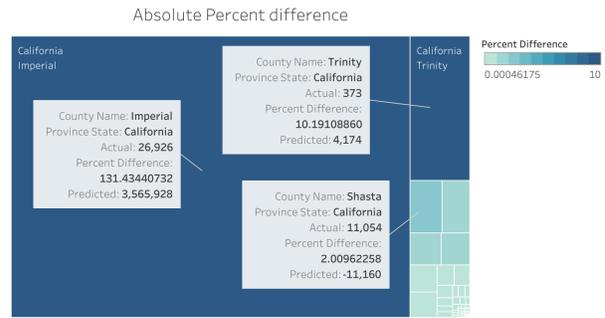


Figure 4: Treemap of Absolute Percent Difference Between Actual Cases and Predicting Cases of California, 3/2/2021

At US country level (figure 6), the model is underestimating the number of cases. At state and county level (Figure 7 and Figure 3), the model performs well predicting the number of cases.

## 5 Discussion

### 5.1 Reason For Inaccurate Predictions on Certain Counties

Fitted Epidemiology Models' predictions are either overestimating or underestimating. The predictions are off because we are isolating the state from neighboring states, county from neighboring counties; there are constant transmissions between neighboring regions.

We want to add another variable into our Epidemiology model, the mobility, how fast are people in an area changing locations, which can be obtained from the mobility data-set at county level. This part will be incorporated in Winter quarter.

Later we tried on a auto supervised learning model to see what we could learn from the performance from the such model. However, because of the limitation of the time series data set, it failed to predict the data of the later time after training on the earlier data set .

From the map above, we can see that the counties we have really "inaccurate" predictions are Imperial County, Amadar County, Shasta County and Trinity County. The missing data for neighboring counties in Arizona for the counties on edges/corners and low population can explain away some of the inaccuracy in predictions for these counties.

Another issue needs to be investigated is the positive or negative percent difference between actual cases

and predicting cases of California on 3/2/2021.

## 5.2 Training Spatial Model For Better Beta and D estimation

Right now, the  $\beta$  and  $D$  are derived from the implementation of Epidemiology Model itself, with out spatial dynamics. However, for future research,  $\beta$  and  $D$  can be learned through Gradient Descent method in the temporal-spatial model.

We would want to minimize the loss function, by setting the derivative of the loss function to zero and solve the function to find the  $\beta$ ,  $D$  and  $c$  to minimize the loss function.

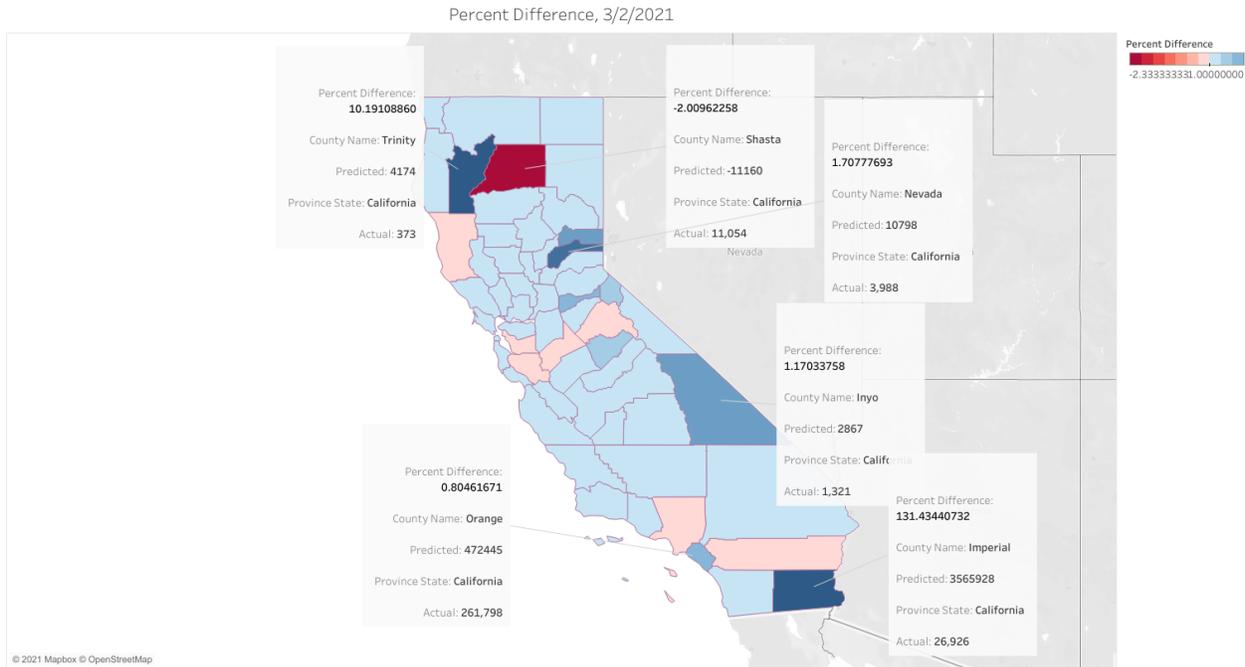


Figure 5: Heatmap of Percent Difference Between Actual Cases and Predicting Cases of California, 3/2/2021

## 6 Appendix

[1] Professor Ma

[2] JHU public GitHub Repository [https://github.com/CSSEGISandData/COVID-19/tree/master/csse\\_covid\\_19\\_data](https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data)

[3] Descarted Lab Mobility Data <https://github.com/descarteslabs/DL-COVID-19>

[4] Lipschitz continuous gradient <https://xingyuzhou.org/blog/notes/Lipschitz-gradient>

### 6.1 Code

```
from numpy import linalg as LA
def get_country(start_days,duration,country = "US"):
    s = [332865671, 332865671, 332865671, 332865671, 332865671, 332865671,
```

```

332865670, 332865670, 332865662, 332865655, 332865632, 332865613,
332865580, 332865503, 332865450, 332865284, 332865168, 332865093,
332864910, 332864537, 332864133, 332863571, 332862804, 332862457,
332861006, 332859167, 332856512, 332852521, 332846735, 332840254,
332831566, 332820683, 332810066, 332798351, 332781931, 332763443,
332744382, 332726361, 332705522, 332678896]
i = [ 16, 16, 16, 16, 16, 16, 17, 17,
      25, 32, 55, 74, 107, 184, 237, 403,
      519, 594, 777, 1150, 1554, 2116, 2883, 3230,
      4681, 6520, 9175, 13166, 18952, 25433, 34121, 45004,
      55621, 67336, 83756, 102244, 121305, 139326, 160165, 186791]
r = [ 5, 5, 5, 5, 6, 6, 6, 7, 8,
      8, 13, 14, 18, 19, 21, 24, 28, 29,
      36, 41, 55, 63, 69, 81, 112, 143, 285,
      358, 480, 606, 699, 859, 1219, 1516, 2165, 2791,
      3581, 5720, 9260, 11426]
p = 332865687
return s,i,r,p

def calculate_gradient(s,i,r,population,beta,epsilon):
    result1 = 0 #continue adding to solve for beta
    result2 = 0 #continue adding to solve for 1/D aka epsilon
    for n in range(len(s)-1):
        result1 += 2*(s[n+1]-s[n]+beta*s[n]*(i[n]/population))*(s[n]*i[n]/population)
        result1 += 2*(i[n+1]-i[n]-beta*s[n]*(i[n]/population) + i[n]*epsilon)*(-s[n]*i[n]/population)

        result2 += 2*(i[n+1]-i[n]+i[n]*epsilon-beta*i[n]*s[n]/population)*(i[n])
        result2 += 2*(r[n+1]-r[n]-i[n]*epsilon)*(-i[n])

    return result1,result2

def calculate(s,i,r,population,learning_rate1,learning_rate2):
    beta = 0.2
    epsilon = 1/14

    loss = 0
    length = len(s)
    betas = []
    ds = []

    for itera in range(1000): # do it for 1000 iterations.

        loss1,loss2 = calculate_gradient(s,i,r,population,beta,epsilon)
        beta_new = beta - learning_rate1* loss1/length #0.001 is the learning rate
        epsilon_new = epsilon - learning_rate2 * loss2/length
        if (beta_new == beta) & (epsilon_new == epsilon):
            print(beta_new)
            print(1/epsilon_new)
            break
        beta = beta_new
        epsilon = epsilon_new
        betas.append(beta)

```

```

        ds.append(1/epsilon)

    return betas,ds
def calculate_hessian(s,i,r,population):
    result_beta_second = 0
    result_epsilon_second = 0
    result_both_second = 0
    for n in range(len(s)-1):
        result_beta_second += 4*(s[n] * i[n]/population) **2
        result_epsilon_second += 4*i[n]
        result_both_second += -2*s[n]*i[n]**2/population
    return result_beta_second/len(s),result_epsilon_second/len(s),result_both_second/len(s)

if __name__ == "__main__":
    #Get 40 days of US data starting on the 30th day since the first case of coronavirus in Wuhan
    s,i,r,p = get_country(30,40)
    top_left,bottom_right,the_other_two = calculate_hessian(s,i,r,p)

    w, v = LA.eigh(np.array([[top_left, the_other_two], [the_other_two, bottom_right]]))
    lip_constant = w[w>0][0]
    learning_rate = 0.1/lip_constant

    iterations = 1000
    betas,ds = calculate(s,i,r,p,learning_rate,iterations)
    plt.plot(betas)
    plt.show()
    plt.plot(ds)
    plt.show()

```

## 6.2 Figures

'us.png'

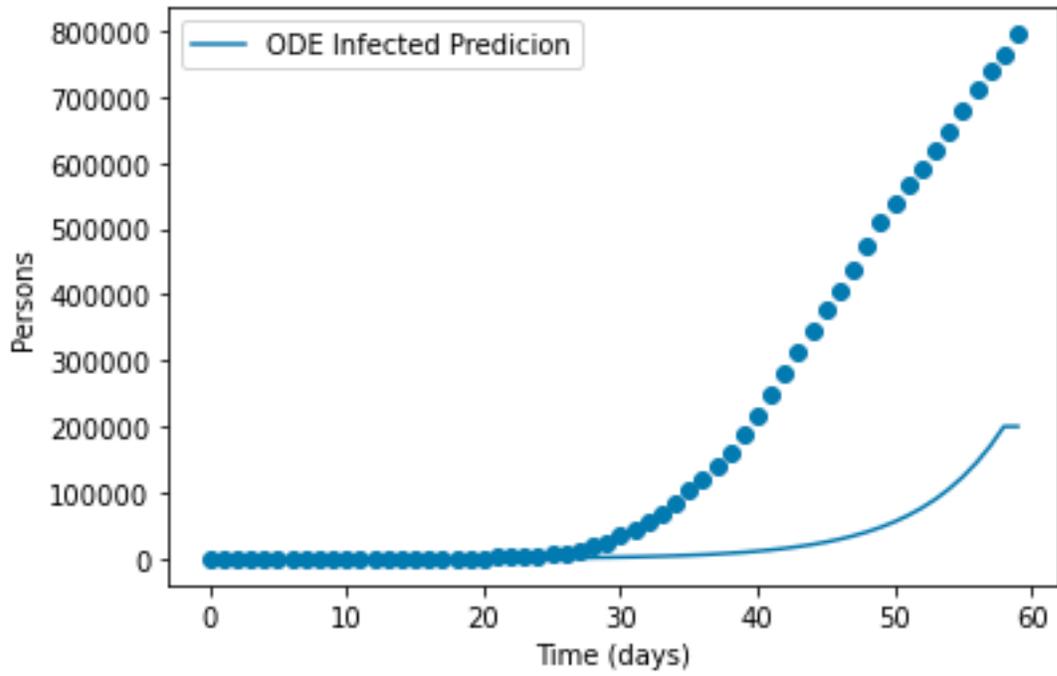


Figure 6: US Country Level, Model Predictions VS Actual Case Numbers

'california.png'

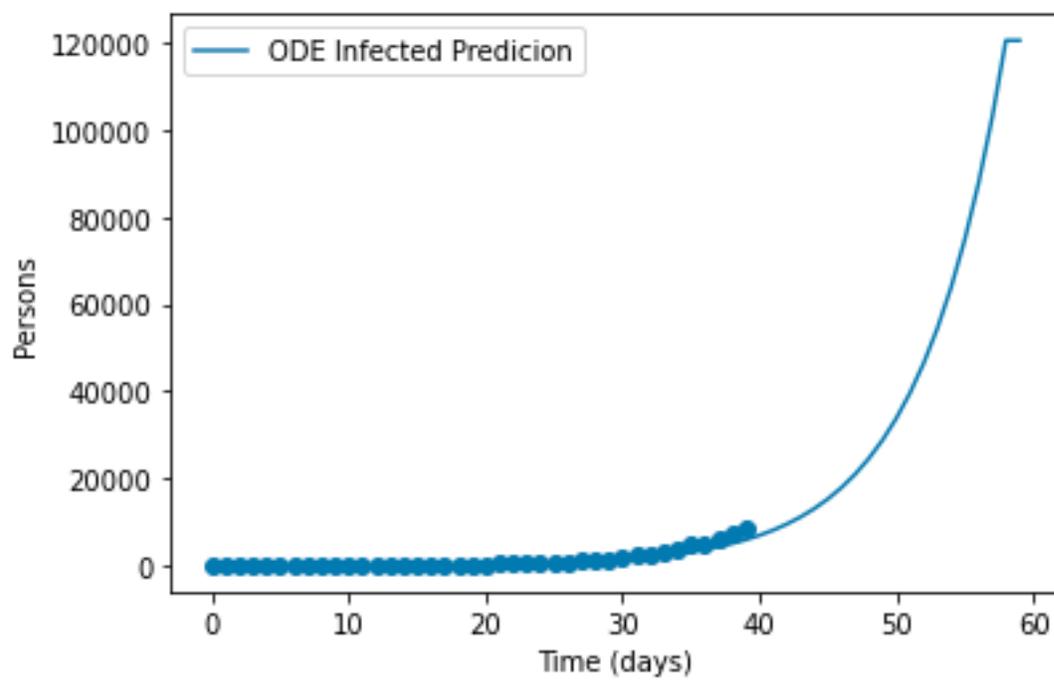


Figure 7: CA State Level 40-60 Days into the Pandemic

'sd.png'

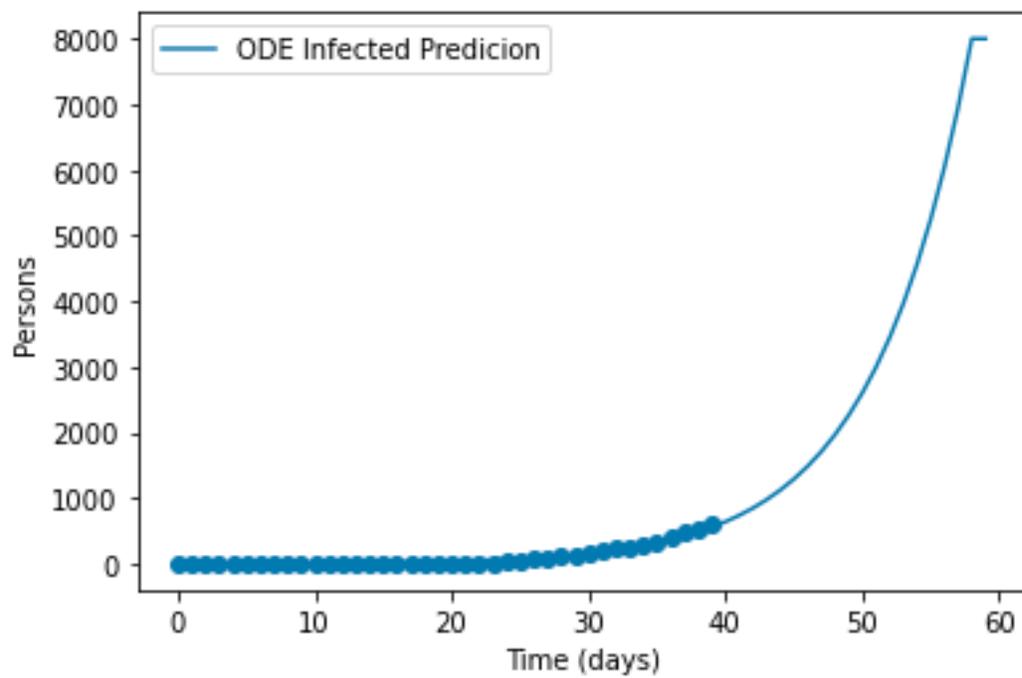


Figure 8: SD CountyLevel 40-60 Days into the Pandemic