Covid-19 project report

Authors: Eric Yang, Shizhe Zeng, Yixuan Hua

Abstract:

There are many things that we still do not understand about the coronavirus pandemic, despite its global and deadly spread. In this project, we constructed models to analyze the characteristics and predict the patterns of covid-19. Our method is to explore the data, and fit different models to find out which one is best suited for the specific question that we are trying to address.

This report tries to address 2 questions: 1) What factors would possibly predict death rates in different counties in the United States? 2) How might historical trends inform our predictions on the number of coronavirus cases in the future?

Since our analysis is mainly focused on the characteristics of the virus, there are little ethical dilemmas we face; the most prominent is perhaps reinforcing existing biases, such as childrens and elderlies are more likely to contract the disease. We cannot claim those are true before the data confirms it. It is important to guard against such tendencies to let ungrounded assumptions affect our model. The way that we went about solving this problem is to avoid human judgement in building the model as much as possible by modelling the data with as little modifications as possible, especially when it comes to selecting features.

1. Data Cleaning:

We did the following steps for data cleaning: 1. we filled the null values of some features with the previous valid value in the column, using the .isna() method with parameter method = 'ffill'. We chose this way of filling null values because the data almost do not have outliers, so it's unlikely that the null values will be filled with a previous outlier value. Also, entries close by belong to the same state, so are likely to have similar values for features. The method of replacing null values with their previous valid value definitely has drawbacks. For example, it would be bad if the previous valid value is of a county that is very different from our current county. But this method is most time-efficient for now.

We also 2 turned some ordinal data (such as "stay at home', ">500 gatherings") into timestamps, so that it would be easier for us to use them as features. We also 3 investigated outliers in terms of mortality rate, and each feature such as "stay at home", and found no significant outliers. Such investigation is necessary because if there does exist any outlier, then we will have to check whether there is some mistake made by people when entering the data. Since we use MSE as our loss function, the loss will be very sensitive to outliers as well. 4 During PCA, we also standardized numerical values in the data by subtracting their means and normalizing their variance to 1. Lastly, 5 we joined the mortality rate to each county according to the state it is in, because the granularities of the two tables are different: one table's granularity is state whereas the other table's is county.

One major problem in this dataset for predicting death rate at the granularity of counties is that the death rate data is at the granularity of state, while almost all other relevant data are by county. This is a problem because we can't go for the coarser granularity of state, since that would make fitting a model very difficult, as there are only 50 states. Therefore, how to infer the death rate of

each county based on that of the state is the biggest problem of this project. In an ideal world, we would have the data based on counties, but for now we've decided to use the death rate of the state to stand in for the county. This isn't a perfect solution by any stretch of the imagination, but this is the most sensible way, since even if not completely identical, the various features that we select are correlated with the state more than any other columns.

2. **The First Question:**

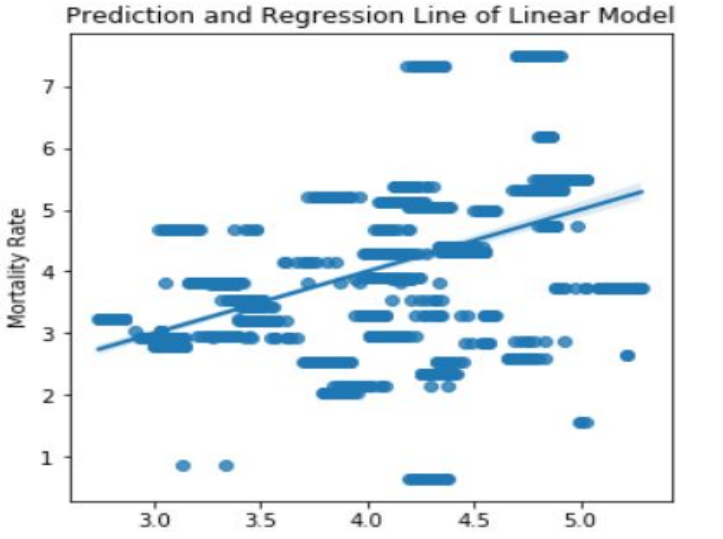
Method and Experiments:

We attempted the following methods: First we tried to select features for the linear model. After dropping some irrelevant columns (such as latitude, county ID), we tried to use PCA to find what accounts for the variances of each county, since all these features are somehow related to the disease. But we don't think it worked, because later we found out that the variance in the medical condition of different states, as represented by the variables, may not be so positively correlated with mortality rate.

As a result, we decided to choose some features that we think should have an impact on mortality rate, and then use cross validation to find out whether those features are actually useful. We used some categorical data and some not-so-obvious numerical features that we obtained from the time series data (such as the date of 1st confirmed case, 1st death, and time when the county reached 100 confirmed). We cleaned the data, with the procedure described in the previous section. We then made a new data frame with all the features, performed train-test split, trained a linear model on the data using SKLearn's linear regression package. We used Mean Squared Error as the loss function, and calculated the loss. We then performed cross validation to delete the features that are not quite useful, which increased the loss. Finally we applied the model on the test set to get the test loss.

Model and Assumptions:

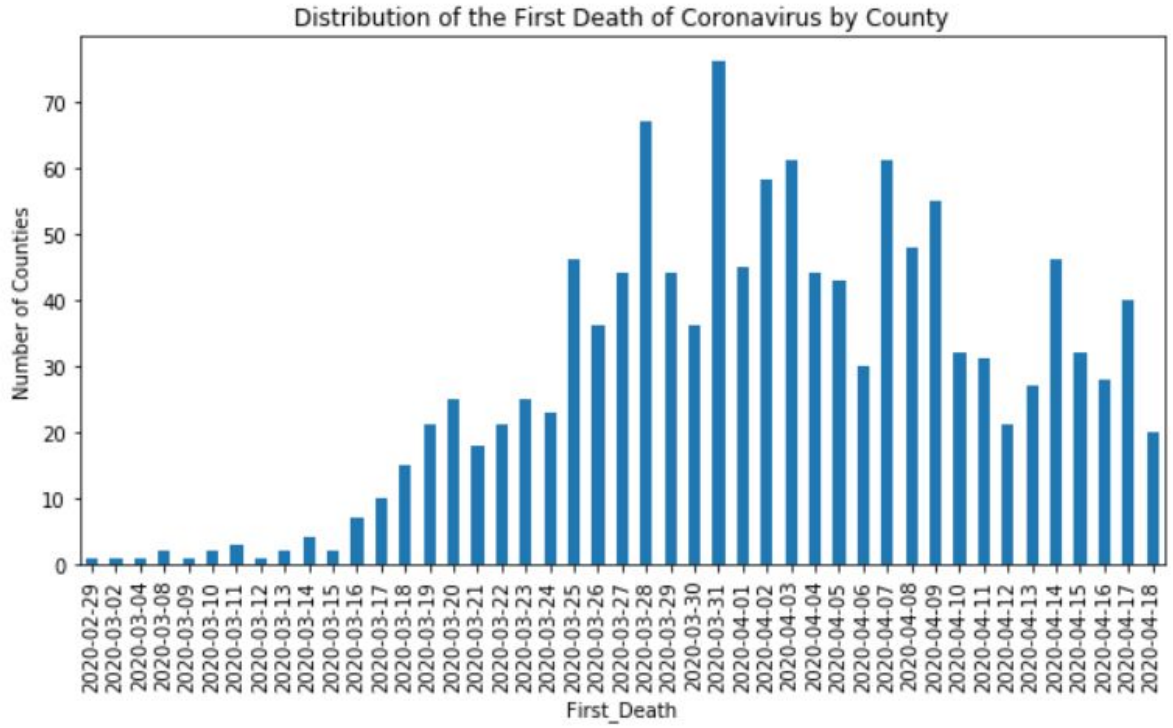
We used a linear model as our model as we assume that mortality rate can be predicted through a linear combination of our features. We did not add in any feature that is the square/square root/cubic term of our current feature, because we did not think it was going to lower our loss.



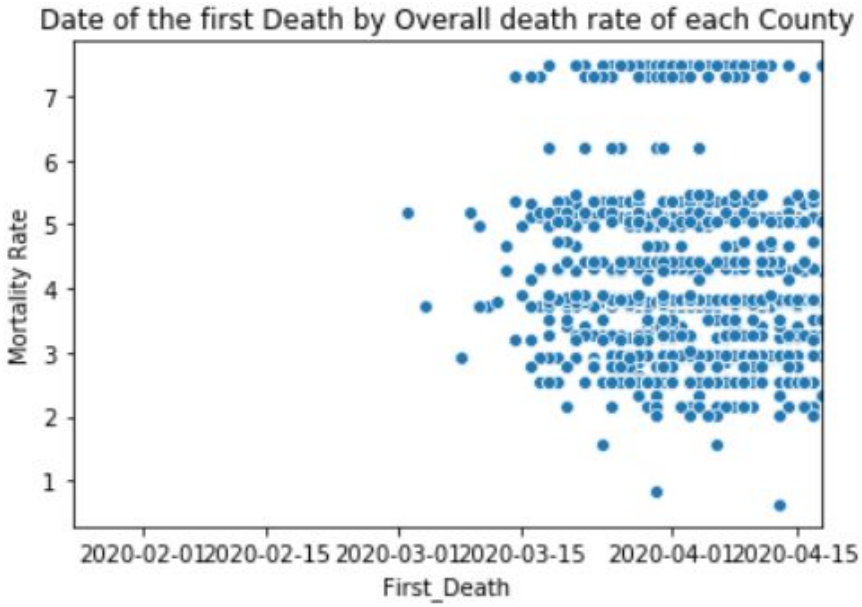
We assumed that the data from the two tables were collected without significant bias. And we also assumed that the mortality of a county is similar to that of the state that the county is in. This is a strong assumption, but we argue that even though counties from the same state can be very different, since counties from the same state should have similar government policy, and we do not have mortality rate of the county-level-granularity, it is sound to use state-level-granularity mortality rate as an approximate.

Analysis and Conclusion:

Our result shows that features like the time that the first confirmed case appears in a county can be a good indicator of mortality rates for counties in the U.S.. We came up with this model that can predict a U.S. county's mortality rate given a county's information (features). We also found that certain features/factors are strongly indicative of mortality rate, while others are not.
Two very interesting features are the first confirmed case and the first death case.



We came up with this idea to utilize the time series we have to make features, and these two features work great in predicting mortality rates. One feature we thought would be useful was the >500 gatherings, but in cross validation it seems like it does not change the loss very much. We guess that it does not work well may be because it overlaps with another feature, as policies like >500 gatherings, staying at home quarantines, and closure of gym/restaurants/school usually are usually co-implemented during a very short period of time. This means that it would be hard to separate their effects from one another, and features representing these policies may be redundant.



Moreover, the scatterplot above shows that the difference between the virus spreading earlier versus later in any given county is negligible. Even worse, none of the highest-death-rate counties were among the first counties to report any death due to the virus. Therefore, the prevention measures taken by the counties either did not take effect soon enough, or they all took similar measures around the same time so as to not show any significant difference in death rate.

The main challenge with our data was that the data is much harder to clean than we imagined. It contained many null values, and we could have done better in replacing the null values with something more useful. Also we found it challenging to make use of both the time series data and the "static" data at the same time, since time is not represented in the first two tables. To solve this problem, we invented some features using time-series data, like the time that the first case appeared, and used them in our analysis.

Since our first simple, linear model did not produce very promising results, we decided to shift to a different kind of model.

3. The Second Question:

The second part focuses on the time-series data sets and tries to build a model that uses historical data to predict future trends. Since we want to predict future cases with existing data, we need a more complex model. One choice is the SIR model, a very well-known model for estimating the spread of pandemics. The SIR model is a type of compartmental model; it separates the population into 3 compartments: Susceptible, Infected, Recovered. In a simple model we define 4 key parameters: β: is the average number of contacts per person per time $\times$ probability of disease transmission in a contact between a susceptible and an infectious subject; D: the number of days someone can carry and spread the disease; γ: the the rate of recovery, intuitively $\gamma = \frac{1}{D}$; and N, the total population, $S(t) + I(t) + R(t) = N$. One other important variable is the basic

reproduction number, R_0. It is the expected number of new infections from a single infection in a population where all subjects are susceptible. We can derive it using: $R_o = \beta \times (1/\gamma) = \beta/\gamma$.

In the Jupyter notebook, there are detailed explanations on how the 3 differential equations: $\frac{dS}{dt} = -\beta I(t)\frac{S(t)}{N}$; $\frac{dI}{dt} = \beta I(t)\frac{S(t)}{N} - \gamma I(t)$; $\frac{dR}{dt} = \gamma I(t)$ were derived and how we add daily deaths into the model. We then project the number of daily cases using Euler's method with a step size equal to 1 day.

Optimizing Model with Gradient Descent

In the previous iteration of the notebook, we have tried using linear models and polynomial models to fit the Daily new cases/deaths curves. Those produce very similar results to the PyTorch example notebook we used in class. (A picture of it is included in the coding zip file). However, the degree 2 polynomial curve does not fit very well with either US daily new cases or daily new deaths. By observing the actual US total daily curve and states' individual curves, we can see that in the beginning of the epidemic, daily new cases increase exponentially. This can also be explained by the SIR model: In the beginning $\frac{S(t)}{N}$ is very close to 1 and relatively unchanged, thus in $\frac{dI}{dt} = \beta I(t)\frac{S(t)}{N} - \gamma I(t)$, $I(t)$ can be approximate by a exponential function ae^{bt+c}. As the pandemic become more serious, rate of increase of infected cases will gradually slow down because of: smaller $\frac{S(t)}{N}$ ratio, lower β because people became more aware of personal hygiene and social distance. In the end, traditional linear or polynomial models fail to capture all the changes in different parameters. Thus, we need a more complex model to fit the curve.

There are several papers detailing the analytical solution to the system of differential equations provided by SIR models. However, they do not provide a simple differentiable solution to I as a function of t; (most of them include integration constant and new parameters that has non linear relationship to β and γ.) Also, we have to define the starting condition S_0, I_0, which can be arbitrary in many cases.

Based on the SIR model, we know that the idealized daily new cases with a constant β is a very symmetrical bell-shaped curve. Thus, we could use a Gaussian function to model the daily new cases as days progress. Interestingly, the statistical model for the cumulative death rate used by Institute of Health Metrics and Evaluation is a parameterized Gaussian error function:

$$D(t;\alpha,\beta,p) = \frac{p}{2}\psi(\alpha(t-\beta)) = \frac{p}{2}(1 + \frac{2}{\sqrt{\pi}}\int_0^{\alpha(t-\beta)} exp(-\tau^2)d\tau)$$

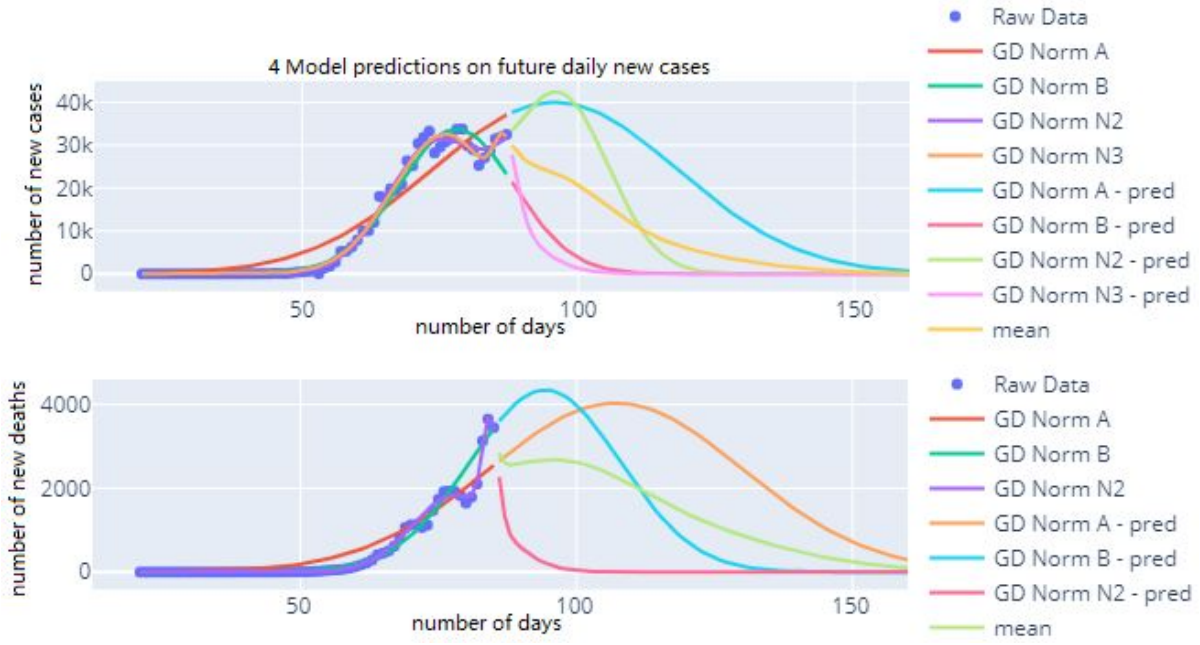
(Where p controls the maximum cumulative death rate at each location, t is the time since death rate exceeded exp(-15), β is a location-specific inflection point (time at which rate of increase of the daily death rate is maximum), and α is a location-specific growth parameter.) [1]

Inspired by this, we build a model that uses standard Gaussian function

$$NewCases(t) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2}$$

[1] "COVID-19 - IHME." https://www.medrxiv.org/content/10.1101/2020.04.21.20074732v1. Accessed 13 May. 2020.

We further simplify it to: $f(x) = ae^{-\frac{(x-\mu)^2}{2\sigma^2}} = ae^{-\frac{(x-b)^2}{c}}$ (where $a = \frac{NewCase_{max}}{\sigma\sqrt{2\pi}}$ and $c = 2\alpha^2$)



Since we are only predicting data based on time-series, there are no feature selections. However, there are still hyperparameters to tune and initial values to set up. After running the normal gradient descent algorithm with a fixed+dynamically-decreasing learning rate (it can't be too small or it will stuck in local minimum) we found out that after a large amount of steps >100k, almost all model will converge to one optimum state with a low MSE loss no matter the initial condition. But it takes a long time even though there is only a small amount of data points and we are optimizing a relatively simple multi-parameter function. Thus, instead of initializing parameters randomly, using educated guesses will help the model converge much faster.

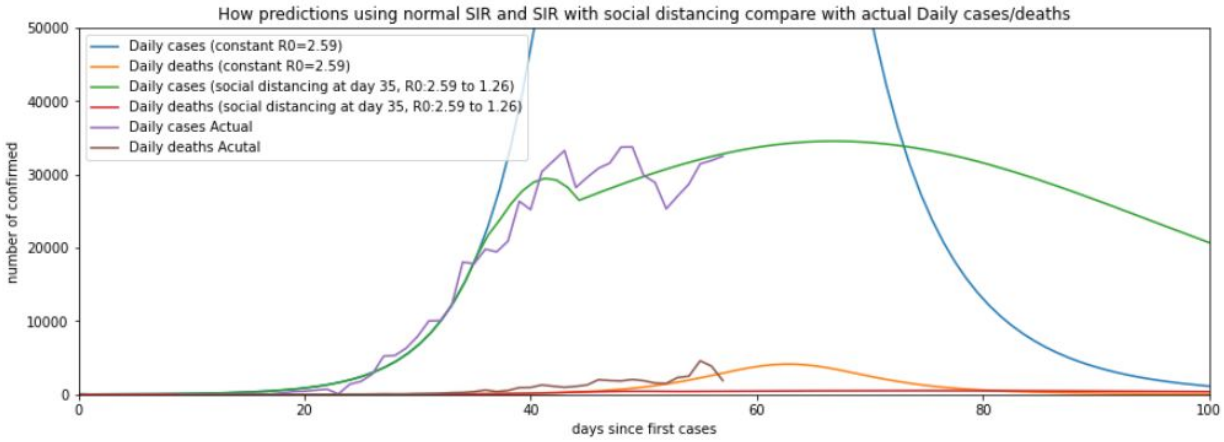
Furthermore, the optimum parameters we are trying to find have a lot of explanatory power over the trajectory of the pandemic. In the simplified model, with only one gaussian function, a measures the value of peak daily new cases. b measures the number of days to reach that peak. c is a measure of how rapid the disease spread and it is related to the value of R_0 in the SIR model. A higher value for c translates to faster disease transmission.

Also, we can use more than one gaussian function to achieve even smaller model loss. But the more gaussian we use to fit the model the more it is prone to overfitting. This is quite apparent in the GD Norm N3 - predict line. Although it has a low (1314k<1552k) loss, it predicts the very unlikely event of the pandemic disappearing in the next 25 days.

There is a lot of challenge when hand tuning the hyperparameters, recording and printing the parameters' 3-dimensional gradient descent steps to see how changing the magnitude of the learning effect the steps, I also tried different types of dynamic learning rate functions and different loss functions such as L1 and custom defined functions to see how it affects the model performance.

Since in the final version the model runs on automatic tuned learning rate "gradient_descent_Adam(...)", the same curve fitting pipeline can be used on time-series data from any of the states or counties (and other countries as well, provided the data), with minimal manual tuning. Thus, with more time and writing space, I would like to run the pipeline on all the states and try to find correlation between each states' descriptive data (such as latitude, longitude, population density, demographics) and the value of c to see if certain features are strongly positively/negatively correlated with the rate of infection.

The model is limited because it is an approximation of the SIR model, which itself over-simplifies the real dynamics of the spreading of the pandemic. Bellow is a modified SIR model with dynamic value for β. As Covid-19 became more and more widespread, states and the Federal government issued Shelter in Place Order. This had a massive impact on the virus transmission rate. As shown in the plot, SIR with constant R_0 vastly exaggerated the daily new cases. Whereas the slightly more sophisticated SIR captures the effect of social distancing and gives a much better prediction.



Overall, the mean of the models produced a prediction that is quite close to the actual data reported after 4/18/2020 considering how rudimentary the model is. But most of it is due to the fact that at 4/18/2020, the pandemic is getting close to its peak allowing the curve to almost trace half of the "bell-shape". If the end-point of the time-series is still during pandamic's early exponential growing phase, the model would produce much wider range of uncertainty.

4. Summary of results, and discussion

In the first questions, we explored different features and found out few such as time of first confirmed case and >500 gatherings are effective in predicting death rate. In the second question, we attempted with a more experimental method. The experience from the past few months has taught us that trying to predict the spread of Covid-19 is very difficult even with state-of-the-art models. With little data, we can only gadge different parameters by comparing the current pandemic with past pandemics. Yet, they all have different characteristics, whether it is death rate, R_0, or incubation period. Performing exploratory data analysis helps us gain deeper insights into the correlation of different variables. Combined with models developed using domain knowledge we can answer crucial questions that otherwise remain unsolved.