Global Minimum Gang COVID-19 model

Kevin Huang, Anish Shenoy, Jae Yoon Kim

Link to project: https://github.com/KevinHuang8/CS156-COVID19

Data Usage and Preprocessing

The only data used in the final model is deaths (both daily and cumulative). Data was smoothed on a per-county basis using a three day moving average, and the data was truncated so that values with 0 cumulative deaths were removed. Counties with no or little data points left (3 or less) are given a special case in the model (described below). See the model description for a better idea of how the data is used.

We also have an unincorporated model that is not part of the final submission, but would have given more time to tune it (see section below). This model uses cases data as well as deaths data, smoothed with a 7 day moving average. This model clusters counties together such that the cases and deaths are combined for each cluster. The clustering method is a geographical clustering within state boundaries (see incorporated model description for more information). Again, describing specific data usage makes little sense without explaining the model, so please refer to the appropriate section for more information.

We attempted to incorporate data other than cases/deaths, especially mobility data, but this never improved our performance with any approach. A likely explanation is that most other types of data are already reflected in the case data. That is, variables like policy changes, mobility, and demographics tend to manifest themselves in the case data to the point where including them as features on top of case data can be redundant and can make the model too complex. In other words, we found that cases at time t can be explained a lot by cases at time $t-k$, but is explained very little by mobility or demographics (especially since mobility remains largely unchanged at the time of the final model submission date). Perhaps these factors had the effect of "seeding" the initial progression of cases, but late time behavior of cases can be determined primarily by looking at cases in the past. The one important exception seems to be rate of testing, because the cases data provided is of course only confirmed cases and not true cases. However, a major challenge is that testing data is not generally available on a county level basis, and is only available for an entire state. We believe that it is fully possible to incorporate this into our model, but we didn't have enough time. We would have liked to examine other types of data more closely, but given limited time, we believed creating and understanding a successful model with just cases/deaths data would be a priority before incorporating more advanced types of data.

Model Description

The model used for our final submission is a least squares curve fit, followed by a Gaussian process to enhance the estimate of the quantiles. We create a separate model for each county.

In particular, we use an exponentially modified gaussian curve, which is given in the following form

$$f(x; A, \mu, \sigma, K) = A\frac{1}{\sigma}\left(\frac{1}{2K}e^{\left(\frac{1}{2K^2}-\frac{(x-\mu)/\sigma}{K}\right)}\mathrm{erfc}(-\frac{(x-\mu)/\sigma-\frac{1}{K}}{\sqrt{2}})\right)$$

where $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$. This curve is meant to fit the daily deaths data. We also sometimes (see training strategy for more details) fit the cumulative deaths data, in which case we use the function

$$F(x; A, \mu, \sigma, K) = A \int_{-\infty}^{x} f(t; 1, \mu, \sigma, K) dt$$

To compute these functions, we use the `scipy.stats.expnorm.pdf` and `scipy.stats.expnorm.cdf` functions, respectively. The least squares fit is performed by `scipy.optimize.curve_fit`. We set bounds of $m \leq A \leq 100m$, $\mu \geq 0$, $\sigma \geq 0$, and $0 \leq K \leq 10$, where m is the maximum daily deaths seen. $K \leq 10$ is an empirically determined hyperparameter.

The reason for using this function is because of two observations: 1) the error function fits the cumulative deaths data to a reasonable degree and 2) daily deaths is asymmetric about the peak.

The error function is the most simplistic model of rapidly increasing deaths at first, followed by slowing deaths. However, we noticed that while deaths do indeed rise rapidly at first, they decrease at a much slower rate after peaking. The error function is constrained to be symmetric about the peak, so it consistently underestimates deaths into the future. Thus, we used the exponentially modified gaussian, as it has the properties of the error function, but is not constrained by symmetry.

The benefits of using a least squares curve fit is that it is very stable in its predictions and simple to implement. The primary drawbacks are that it is incapable of incorporating data from multiple counties or other data sources. Furthermore, it will fail if there is ever a "second wave" of increasing deaths after the initial peak. However, at the time of model submission, no county has experienced this phenomenon.

To obtain accurate quantile predictions, we take the least squares curve fit, and estimate the distribution of the noise of the data based on the residuals, assuming a constant distribution over time (see training strategy for more details). We then hallucinate data points into the future using our least squares curve fit and the estimated noise, and train a Gaussian process on the hallucinated data points. This Gaussian process uses our least squares curve fit as the prior mean and a radial basis function kernel. The quantiles were then obtained by sampling the Gaussian process, and taking the respective percentiles.

For counties with no deaths at all, we simply predict zeros indefinitely. For counties with few deaths, we predict a constant value, equal to the average of daily deaths in the past.

Training Strategy

We first split up the data into training and validation sets by reserving two weeks from the end for validation. For each county, we then train a curve fit model, on BOTH the daily deaths data, and the cumulative deaths data (both smoothed with a 3 day moving average). We then compare the two models on the validation data, and pick the one with better mean squared error.

We then train on the whole data. For each county, we train a curve fit model, using either the cumulative or the non-cumulative model, depending on which performed better in the previous step. If we train a cumulative model, we difference the predictions, so we are always prediction daily deaths.

For each county, we then obtain the residuals between the *non-smoothed* daily deaths data and the fitted curve. We assume that the noise in our data is normally distributed with variance equal to the variance of the residuals (there are better ways to estimate the noise distribution, but this is the fastest). We then assume that daily deaths d is equal to the following formula:

$d(x) = f^*(x) + \epsilon$ where f^* is our curve fit, and ϵ is the noise given by the estimated residual distribution. We sample d according to this formula for $x = 0$ up until the future horizon we want to predict for. We then fit a Gaussian process with these hallucinated data points, and then sample the Gaussian process to obtain quantile estimates. We repeat this process 3 times for each county, and take the average prediction.

Note that this is not a 2-tiered approach that trains on the residuals, as we only use the residuals to estimate the variance in the original data. Our mean always remains the curve fit.

We performed validation by taking two weeks from the end and excluding it from training. Note that we denote this the "test set" and not the "validation set" mentioned at the very beginning. The test set is never touched during training, and the validation set used for comparing cumulative vs. non-cumulative models excludes an extra two weeks on top of the two weeks excluded for the test set.

The only meaningful hyperparameters are the upper bound on the parameter K, and the size of the moving average used to smooth the data, both of which were selected empirically. Given more time, we would liked to have determined these more analytically using a Bayesian optimization approach.

Model Robustness

We will look at the past performance of our final model, trained on data up until 5/14. The counties with the largest pinball loss are:

1. LA County (06037): 9.24
2. NYC (36061): 7.06
3. Chicago (17031): 6.87
4. Philadelphia (42101): 5.39
5. Detroit (26163): 4.90

However, the counties the the largest average percent error are:

1. St Louis (29189): 24.1%
2. Oakland County, MI (01293): 22.9%
3. McKinley County, NM (35031): 21.1%
4. Detroit (26163): 20.0%
5. Phoenix (04013): 18.1%

No county has predictions that wildly diverged from reality. This is because of the nature of our model; past the peak, the curve is always decreasing, so our predictions never diverge, and no county has seen daily deaths rise after an initial peak, if it exists. For counties without an initial peak, our model does not perform as well, since it must guess when the peak will occur. However, these counties all have low population, so they do not contribute much to our overall loss. Our model also performs relatively poorly on areas with wide variances in daily deaths, such as LA, Chicago, and St. Louis. For a lot of counties, this variance seems to be systemic and has to do with how deaths are reported, or with predictable trends like weekend effects. For example, some counties seem to report cases in bulk at certain points in time, not when the deaths actually happen, resulting in large spikes and dips in the deaths data. This is especially notable for Providence, RI. However, we do not attempt to predict it in our model, we only attempt to predict a smoothed moving average, and try to encapsulate the variance in our error bars.

Overall, our model is very stable, though it will entirely fail in the case of a second wave.

Unincorporated Model

We will briefly describe a much more complex strategy we have been working on that incorporates case data as well. It was a promising strategy, but unfortunately did not have enough time to tune it to the point where we could be confident in its performance.

This strategy attempts to predict deaths at time t using cases at time $t-k$ (both deaths and cases in this case are smoothed with a 7 day moving average), which will then be blended with the results from the main model. The intention was for this strategy to pick up on the effects of an uptick in cases to make the model resilient to a surge in cases.

We first cluster counties in the same state together using a nearest neighbors method (implemented as a simple BFS), and aggregate cases and deaths. This is because we noticed that the relationship between cases in the past and present deaths for nearby counties is generally very similar.

For each cluster, we determine the optimal lag k to use by taking the k with the highest correlation coefficient r between deaths at time t and cases at time $t-k$. We then perform a linear least squares fit between past cases and present deaths, and use this as the prior of a Gaussian process with an RBF kernel. We sample this Gaussian process to obtain quantile estimates.

For each county, we then compare the predicted deaths in the past using the Gaussian process for the cluster the county is contained in to the true deaths in the past. This is done to determine any differences between the county and the cluster as a whole. We then adjust the quantile estimates accordingly.

For inference into the future, we potentially need cases in the future, so we use a exponential gaussian curve fit model (our main model) to predict future cases. This should be replaced with a different model ideally, but we just went with what we had.

Given more time, we would have like to develop another model that would have combined the predictions from this model with our curve fit model. In addition, this model could have incorporated other types of data, especially testing data, so we could get a more accurate measure of true cases. We believed that including this model would have improved our predictions, though we didn't have enough time to tune everything.

Failed Models

The neural network approach we first used had underwhelming performance. We used a LSTM with attention, trained on the cumulative deaths data with second order differencing. It did not have terrible performance, but was not as robust or accurate as simple curve fitting models. The main issue seemed to have been the fact that the LSTM required stationary data to train well, but the second order differenced data (which was needed for stationarity) had so much noise that it dominated the signal, so the LSTM was unable to make stable predictions.

Model Sensitivity

The noise in the data is explicitly baked into how we calculate our error bars. This is because we assumed that the variance in the residuals between the true data and our curve fit is the variance of the true noise in our data (both stochastic and deterministic noise), where the noise is normally distributed.

Let us take another measure of the distribution of the noise by taking the difference between the unsmoothed daily deaths data and a 7 day moving average smoothed data. We assume that the standard deviation of these points is the true standard deviation of the noise. If we do this with the 50 counties with the most deaths, we find that on average, the 90th-10th percentile range is 2.143 standard deviations (as measured by the above technique). If we assume that daily deaths are normally distributed, then we would expect that the 90th-10th percentile range is about 2.564 standard deviations. Thus, it seems that might be slightly underestimating the level of noise, and increasing the error bars might improve the loss.

However, most of the error, like we stated in the robustness section, seems to stem from the fact that there are systemic reporting quirks in many counties, so deaths will spike up and down. That is, a lot of the noise seems to be deterministic noise in that it is possible to predict, but our model cannot capture it.

Long-Term Predictions

We calculate the error of our model across various one month windows in the past:

Last date of data model was trained on	Pinball Loss	MSE
4/22	0.307	51.68
4/29	0.190	11.57
5/6	0.118	3.05

As we see, our model performs better later in the progression of the pandemic. This is likely because, as stated above, our model works better when the first peak of deaths has already been reached. With later dates, more counties are likelier to have their peak reached, which results in better performance. Also, since our fitted curve is monotone decreasing after the peak, the loss naturally tends to decrease as daily deaths decreases. We expect that this trend would not be as prevalent had we incorporated our second model that used case data, as that would allow us to predict when a peak will occur. Again, we would note that if there is a second wave in the future, our current model would totally collapse.

International Comparisons

Since our main model only uses deaths, it is simply to apply it to international predictions. Applying it Italy, training on data up until 5/24, we find that the average absolute percent error for a horizon of 7 days is 20.4%, which is right in line with counties in the US. This is to be expected, as our model is stable enough that our curve fit should generalize to areas of different demographics and policies. We expect similar results