Appendix B: CurveFit Tool and Analyses

Abstract

This Appendix gives details for the `CurveFit` program and related analyses, including age standardization, and peak detection, which are used together with the tool to obtain the estimates for the Covid-19 death rates in the public-facing tool https://covid19.healthdata.org/. The report includes methods used to create the initial estimates, as well as updates that have been developed over the last three weeks. The tool allows multiple functional forms, covariates, link functions, and prior specifications, that can be used as we learn more about Covid-19. A Gaussian form for daily deaths remains the workhorse functional form used thus far. To fit distributions of daily deaths, which exhibit asymmetry and flat peaks across locations, we fit a linear combination of Gaussian atoms to the data. Uncertainty is estimated in all cases using a model-agnostic predictive validity framework, also detailed in the report. The mathematical methods are open source, and the repository cited in the introduction is updated as the work continues to evolve.

1. Introduction

Overview. The `CurveFit` package, available at https://github.com/ihmeuw-msca/CurveFit, is used by IHME to estimate and forecast deaths across locations[1]. General changes in data, covariates, and models are described on the main website[2] as the approach evolves.

The forecasts for Covid-19 deaths and equipment need assume that:

(1) All social distancing measures that are in place will stay in place.

(2) Any remaining restrictions will be put in place within a fixed number of days.

The time before the remaining social distancing measures are to be implemented was assumed to be 7 days prior to April 17 forecasts, and 21 days for forecast published on April 17 and afterwards.

CurveFit Model. `CurveFit` supports parametrized curves that can be fit to data, modeling parameters using covariates, and post-processing, such as fitting linear combinations of CurfeFit models. We focus on parametric and semi-parametric inference (in contrast to fully nonparametric inference, e.g. fitting tools with splines [10]) for several reasons:

- Parametric functions capture key signals from noisy data due to simple parametrization.
- Parameters are interpretable, and can be modeled using covariates in a transparent way.
- Parametric forms allow for more stable inversion approaches, for current and future work.
- Parametric functions impose rigid assumptions that make forecasting more stable.

Roadmap. The Appendix proceeds as follows. Age-standardization, an important pre-processing step done for each forecasted location before running `CurveFit`, is described in Section 2. For the Covid model, we considered sigmoidal shapes, described in Section 3. Assumptions on noise and relationships between locations are specified through the statistical model, discussed in Section 4. Covariate definitions for original and updated analyses are given in Section 5. Assumptions and expert knowledge can be communicated to the model through priors and constraints, described in Section 6. All estimation is carried out using an optimization procedure, described in Section 7. The extended model that fits a constrained linear combination of Gaussian atoms discovered by fitting the basic `CurveFit` model is given in Section 8. Posterior uncertainty is estimated from the fits using a prediction validity framework described in Section 9. Automatic peak detection used to get a set of likely peaked locations for further expert vetting using splines with shape constraints is detailed in Section 10. Current settings used to obtain fits are summarized in Section 11.

2. Age Standardization

In an effort to control for the confounding effect of age structure variation across the geographic units for which we estimate COVID-19 deaths, we run separate model pipelines for each location,

[1] https://covid19.healthdata.org/projections
[2] http://www.healthdata.org/covid/updates

standardizing all data to that location's population age structure. The key pre-processing step before the analysis is to convert the reported cumulative deaths in our dataset into death rates using the most recent available population data from the Global Burden of Disease 2019 study.

We use the average age pattern of COVID-19 mortality rates in 10-year age bands up to a terminal group 80+ based on data from Hubei, Italy, Republic of Korea, and the United States as a reference mortality rate by age m_a^r. We then derive an implied mortality rate m_l^i using those data and the age-specific population of each location in the model dataset $p_{a,l}$.

$$m_l^i = \sum_{a=[0-9]}^{[80+]} \frac{m_a^r \times p_{a,l}}{p_l}$$

We can then adjust the reference age pattern by the ratio of the observed mortality rate on a given location-day $m_{l,d}^o$ to the implied mortality rate to produce a series of age-specific mortality rates $m_{a,l,d}$ representative of each datapoint.

$$\{m_{a,l,d}\}_{a=[0-9]}^{[80+]} = \{m_a^r\}_{a=[0-9]}^{[80+]} \frac{m_{l,d}^o}{m_l^i}$$

Lastly, we apply the population structure in the model location p_{m_l} to the age-specific mortality rates created from each data point, resulting in an age-standardized mortality rate $m_{l,d}^{as}$.

$$m_{l,d}^{as} = \frac{\sum_{a=[0-9]}^{[80+]} m_{a,l,d} \times p_{a,m_l}}{p_{m,l}}$$

The natural log of the age-standardized mortality rate is then used as input data to the CurveFit model.

3. Functional Form for Covid-19

We considered several functional forms to model the death rate of the Covid-19 virus. Based on currently available data, the log rate starts slowly, increases quickly, and then flattens out again as either social distancing or saturation goes into effect. This is the classic sigmoid shape. We first tried building the analysis using the sigmoidal function

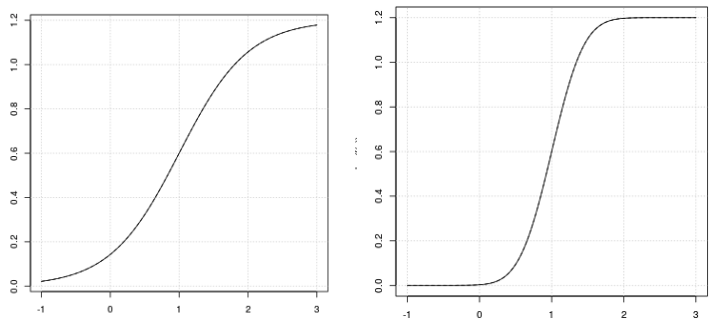

Figure 1. Expit function $\widetilde{D}$ (left) and ERF function D (right). The ERF function fits the available Covid-19 data better than Expit.

$$\widetilde{D}(t;\alpha,\beta,p) = \frac{p}{1+\exp(-\alpha(t-\beta))}$$

where p controls the level, β the shift, and α the growth. **Here and below, we refer to fundamental quantities, here p, β, α as parameters.**

We then discovered that the ERF error function provided a better fit to the data:

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

`CurveFit` allows the user to specify an arbitrary parameterized functional form, so that other models could be considered as more data becomes available. We can fit in four spaces:

- Log space: log (data) vs. log(D)

- Linear space: data vs. D
- Derivative of log space: increments of log data vs. derivative of $\log(D)$
- Derivative of linear space: increments of data vs. derivative of D.

For the D functional form, the three parameters are:

- Level: p controls the maximum asymptotic level that the rate can reach
- Slope: α controls the speed of the infection
- Inflection: β is the time at which the rate of change of D is maximal.

These interpretations are clear from the following derivative computations:

Logistic Function.

$$\widetilde{D}(t) = \frac{p}{1+\exp(-\alpha(t-\beta))} = p(1+\exp(-\alpha(t-\beta)))^{-1}$$
$$\widetilde{D}'(t) = p\alpha(1+\exp(-\alpha(t-\beta)))^{-2}\exp(-\alpha(t-\beta)) = \frac{p\alpha}{\exp(\alpha(t-\beta))+2+\exp(-\alpha(t-\beta))}$$
$$\widetilde{D}''(t) = \frac{-p\alpha^2\left(\exp(\alpha(t-\beta)-\exp(-\alpha(t-\beta))\right)}{(\exp(\alpha(t-\beta))+2+\exp(-\alpha(t-\beta)))^2}$$

It is clear that $\widetilde{D}'(t)$ is maximized at $t=\beta$, since the numerator of $\widetilde{D}''$ is then equal to 0, that is the infection point occurs at $t=\beta$. Plugging in, the maximum value of $\widetilde{D}'$ is given by

$$\widetilde{D}'(t)_{\max} = \frac{p\alpha}{4}.$$

We can also obtain a simple expression for $D''(t)$ at $t=0$:

$$\widetilde{D}''(0) = p\alpha^2\frac{(\exp(\alpha\beta)-\exp(-\alpha\beta))}{(\exp(-\alpha\beta))+2+\exp(\alpha\beta))^2} \tag{1}$$

ERF Function.

$$D(t) = \frac{p}{2}\left(1+\frac{2}{\sqrt{\pi}}\int_0^{\alpha(t-\beta)}\exp\left(-\tau^2\right)d\tau\right)$$
$$D'(t) = \frac{p\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2(t-\beta)^2\right)$$
$$D''(t) = \frac{-2p\alpha^3}{\sqrt{\pi}}\exp\left(-(\alpha^2(t-\beta)^2\right)(t-\beta)$$

It is clear that $D'(t)$ is maximized at $t=\beta$, since at that value $D''(t)=0$. Plugging in, the maximum value of $D'(t)$ is given by

$$D'(t)_{\max} = \frac{p\alpha}{\sqrt{\pi}}.$$

For the ERF function, we also have a simple expression for the rate of change of daily deaths at $t=0$:

$$D''(0) = \frac{2p\alpha^2}{\sqrt{\pi}}(\alpha\beta)\exp\left(-(\alpha\beta)^2\right) \tag{2}$$

In both functional forms, the maximal D' expressions are proportional to $p\alpha$, and the rates of changes D'' at $t=0$ (and at other specified times) are strongly dependent on the quantity $\alpha\beta$.

Asymmetric Extensions. We also considered asymmetric forms, such as a switched Gaussian devised by one of the team members. For the ongoing analyses, we still use a symmetric form, but fit to data using a linear combination of the inferred peaks described in Section 8. This approach has proven robust, while also fitting a variety of asymmetric data without relying on a particular functional form with additional parameters.

To capture variation across location, we have to model the relationships of parameters using covariates and random effects. These specifications are given in the next section.

4. Statistical Specification

Statistical assumptions link parameters together across locations. Statistical models introduce variables that can be inferred to describe these relationships. `CurveFit` allows any parameter to be specified using both a link function and covariates using the generalized linear modeling framework [7]:

$$\text{parameter} = \text{LinkFunction}(\text{covariate} * \text{multiplier} + \text{random effect}).$$

The covariate value is provided by the user (for example, a measure of social distancing), while the multiplier and random effect are both variables that are solved for using an optimization procedure from the data. **Here and below, we use 'variables' to refer to quantities solved for by an algorithm**. We use the word 'parameters' only when talking about (α, β, p).

For the Covid-19 model, there are two link functions:

- identity for modeling the β parameter, and
- exponential function to ensure that α and p parameters are positive.

The ability to parametrize by covariates is a key functionality of the model. For example, the only covariate used in the death rate model for the current estimate is based on the duration between a threshold of the rate and social distancing policy, and this covariate drives the inference of the covariate multiplier for the inflection point or the level in the models we consider for the analysis. As more data becomes available, `CurveFit` can be used to incorporate additional understanding to further link the covariates.

To finish the specification, we give an important modeling example that is used for current estimates. The covariate links the inflection points β_l across locations l. The observation model is

$$\log(\text{cumulative death rate in location } l \text{ at time } t) = \log(D(t; \alpha_l, \beta_l, p_l)) + \text{error}_{l,t}$$

and the remaining specification is

$$\begin{aligned} \alpha^l &= \exp(\mu_\alpha + u_\alpha^l) \\ \beta_l &= (\mu_\gamma + u_\gamma^l)\text{Covariate}^l \\ p^l &= \exp(\mu_p + u_p^l) \end{aligned} \tag{3}$$

In this example,

- μ_α and μ_p are intercepts (in log space) that capture average behavior of parameters α^l and p^l across locations
- u_α^l and u_p^l are random effects that multiplicatively adjust $\exp(\mu_\alpha)$ and $\exp(\mu_p)$ to each location
- μ_γ is the average covariate multiplier that controls the peak β
- u_γ^l are random effects on slope that adjust the covariate multiplier to each location.

5. Covariate Definitions

The covariate in the `CurveFit` model (3) is very important in being able to predict the peak. The information used to construct the covariate has evolved between the initial posting of the model and the current iteration, and the procedure is briefly described here. The procedure describes creation of multiple covariates by treating the available information differently, to create a set of models in the model pool that are then ensembled to create the final estimates as discussed in Section 11.

5.1. Social Distancing Covariates Prior to Social Mobility Data

Before social distancing data was available and had been processed by the team, government mandates across locations were used to construct the covariate to capture social distancing (see Supplementary Information). Specifically, covariates of days with expected exponential growth in the cumulative death rate were created using information on the number of days after the death rate exceeded 0.31 per million to the day when 4 different social distancing measures were mandated by local and national governments: school closures, non-essential business closures, stay-at-home recommendations, and severe local travel restrictions including public transport closures. Three different weighting schemes to create covariates were considered:

1. Days with 1 measure were counted as 0.67 equivalents, days with 2 measures as 0.334 equivalents and with 3 or 4 measures as 0;

2. Days with 1 measure were counted as 0.86, 2 measures as 0.57, and 3 or 4 as 0
3. Days with 1 or 2 measures are counted fully, and 3 or 4 counted as 0.

For locations that have not yet implemented all of the closure measures, the forecasts assumed that the remaining measures would be put in place within 1 week of the data of analysis. This lag between reaching a threshold death rate and implementing more aggressive social distancing was combined with the observed period of exponential growth in the cumulative death rate seen in Wuhan after Level 4 social distancing was implemented, adjusted for the median time from incidence to death. For ease of interpretation of statistical coefficients, this covariate was normalized so the value for Wuhan was 1.

5.2. Using Social Mobility Data

The model run on April 17 and future updates use population-level mobility data to better reflect how populations are changing their behavior once distancing mandates are implemented. That means we now inform our model predictions by including information on how populations are responding to different distancing measures.

We use social mobility data from Descartes Labs[3], SafeGraph[4], and Google (via their COVID-19 Community Mobility Reports)[5] in relation to each type of distancing policy implemented. All three mobility datasets are available for the US, while the Google mobility dataset is the only one that includes European countries.

Each dataset is analyzed separately to estimate the percentage reduction in mobility associated with each of our six social distancing measures. We then use these estimates as weights to construct a single covariate for predicting the epidemic peak in each location, see Table 1. We produce three distinct versions of the social distancing covariate (i.e., one based on data from Descartes Lab, one from SafeGraph, and one from Google). We run the COVID-19 death model for each of the three versions of the social distancing covariate and then ensemble them into a single set of predictions.

Table 1. Mobility weights

	Any Gathering Restrictions	Stay at Home	Ed. Fac. Closed	Any Business Closures	Non-ess. Serv. Closed
Descrates	0.129	0.206	0.274	0.212	0.178
Google	0.222	0.081	0.37	0.176	0.151
Safegraph	0.206	0.277	0.201	0.141	0.175

We use "Any gathering restrictions" as an incremental implementation of "People instructed to stay at home", so the full mandate is the sum of weights in the first two columns of Table 1. The same is true of "Any business closures" and "Non-essential services closed". Using these values, we determine the weighted average of days without each mandate. For example, when using Descrates data, the weighted average for a given location using Table 1 is computed as below:
0.129 * (Days without any gathering restrictions) + 0.206 * (Days without a stay home order) + 0.274 * (Days with open educational facilities) + 0.212* (Days without any business closure) + 0.178 *(Days without a non-essential services closed order).

As done in Section 5.1, this composite measure is then combined with the empirical closure to peak duration (21 days), and normalized based on the Wuhan value (so Wuhan has value 1). Since switching to these weights, we have also revised the duration of time before unimplemented mandates are presumed to be in place from 1 week to 3 weeks in the future from the day at which the forecast is obtained.

6. Specifying Priors and Constraints

The `CurveFit` tool lets the user specify prior knowledge using two interfaces: Bayesian priors and constraints. Both types of information can be used to inform estimation of all parameters and covariate multipliers. In the sections below we discuss simple priors, box constraints, and functional priors.

6.1. Simple priors

`CurveFit` assumes that prior distributions are Gaussian $N(\mu, \sigma^2)$, where the parameter μ encodes the prior belief, while σ^2 specifies confidence in this belief.

[3] https://github.com/descarteslabs/DL-COVID-19
[4] https://www.safegraph.com/dashboard/covid19-commerce-patterns?is=5e8b94eac6a05447bd786ae9
[5] https://www.google.com/covid19/mobility/

6.2. Box constraints

Constraints are assumed to be simple bound constraints, that is, we can specify

$$\text{lower bound} \leq \text{parameter} \leq \text{upper bound}$$

for any parameter we wish to infer. Since the functional form D is highly nonlinear, constraints are very useful in stabilizing the numerical solution of the inference problem and communicating model assumptions about parameters in a simple way. Constraints guarantee that parameters will stay in a certain range, but do not prescribe any particular value in that range.

6.3. Functional priors

The behavior of nonlinear curves often depends on coupled relationships between parameters. For example, rates of change of daily deaths D'' depend on all three parameters (p, α, β), see (1) and (2), and strongly depend on the quantity $\alpha\beta$. `CurveFit` therefore allows functional priors, which for the logistic functions can be written as

$$f(\alpha, \beta, p) \sim N(\mu, \sigma^2).$$

These priors can be used when the generalizable quantity (i.e. information we learn from locations with a lot of data) is a function of the modeled parameters.

7. Optimization Procedure

The final optimization problem includes the GLM specifications such as (3), along with Gaussian priors (simple and functional) and bound constraints. The fitting problem in the current version of `CurveFit` is thus a bound-constrained nonlinear least squares problem. To solve this optimization problem, we use the L-BFGS-B algorithm [11], implemented in `SciPy`[6].

The L-BFGS-B algorithm requires derivatives of the objective function. We use numerical differentiation, implemented using the complex step method, to compute these derivatives for any user-specified functional form [5]. Complex step is a simple variant of Algorithmic Differentiation (AD) [2]. More sophisticated packages are being tested, but if adopted will impact speed of the method rather than results.

Since the curves are highly nonlinear, the nonlinear least squares problem is highly nonconvex, and therefore initialization is important. When fitting a joint model for multiple locations, we initialize values of the random effects parameters to their location-specific fits, and then run the full optimization model as specified in Section 4 from this starting point.

8. Curve Fitting Extension Using Gaussians Atoms

As we see more and more data across locations, it is clear that while some peaks follow the classic Gaussian shape in daily deaths, many do not. Some peaks are wider, some trajectories are asymmetric, and overall there is a fair amount of variation in the shape of the curves we see directly in the data.

To balance model flexibility (fitting data) with generalizability (forecasting potential epidemic trajectories), we use a semi-parametric modeling framework, building on the basic `CurveFit` result. The steps are as follows:

- We fit a particular `CurveFit` model to a given location using the social distancing covariate, to fit its γ multiplier, α, and p (see (3)). This gives the atom specification for the next step.
- Given the atom, we use a semi-parametric fit of staggered atoms to data. Specifically, we consider a basis of staggered atoms 13 days, with peaks 2 days apart, centered at the inferred peak from step 1. We fit the weights to the data as described below.

Fitting procedure. Given a set of atomic functions of time $f_i(t)$, and all observations y_t for a given location, we fit the following model:

$$y_t = \sum_{i=1}^{13} w_i f_i(t) + \epsilon.$$

[6] https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html

The resulting models generalize the basic model used so far and better capture the signals in the data – in particular the fitted combinations of curves can be asymmetric, and exhibit flatter regions. Overall the approach better captures the variation in the epidemic trends that we see. At the same time, the extended model is can still be used to forecast into the future just as in the original single atom case.

We want to fit the data as a non-negative combination of atoms. We also put upper bound constraints of 1 on each weight. The full fitting problem is given by

$$\min_{\{0 \le w_i \le 1\}} \sum_t \left(y_t - \sum_{i=1}^{13} w_i f_i(t) \right)^2 . \tag{4}$$

Problem (4) is a bound-constrained linear least squares problem, in particular convex, and easy to solve. It is analogous to a spline, except that the atoms are highly structured – simple replicates of the peak inferred from the data. Since (4) is a least squares problem with bound constraints, we also use the L-BFGS-B routine to solve it.

Uncertainty for any model fit (including the basic fit and the extension) is computed using the predictive validity framework, described in the next section.

9. Uncertainty Quantification

`CurveFit` provides draws – random realizations of the mean function – for individual locations used in the model estimation. Location-specific samples then inform aggregate uncertainty of downstream estimates. To make these draws, `CurveFit` can use sampling based on either approximated model-based uncertainty, or based on predictive validity. While uncertainty for the initial forecasts (updated March 30-April 1st) were made using model-based uncertainty (Section 9.1), the uncertainty for the forecasts on April 5th were computed via the predictive validity framework (Section 9.2).

9.1. Model-based uncertainty

We partition the uncertainty as coming from two sources: fixed effects and random effects. Fixed effects in the model are average parameters across locations, and covariate multipliers. Random effects are specific to location. Estimates of uncertainty for both pieces of the model come from asymptotic statistical approximations (Fisher information) together with the likelihood.

Fixed Effects. For any estimator obtained by solving a nonlinear least squares problem

$$\hat{\theta} = \arg\min_{\theta} := \frac{1}{2\sigma^2} \|y - f(\theta; X)\|^2_{\Sigma^{-1}}$$

we can approximate posterior covariance using the inverse of the Fisher information matrix:

$$\mathcal{I}(\theta) = V[\nabla \mathcal{M}(\theta)] = V[J_\theta^T \Sigma^{-1}(f(\theta; X) - y)] = J_\theta^T \Sigma^{-1} J_\theta$$

where

$$J_{\hat{\theta}} := \nabla_\theta f(\theta; X)|_{\theta=\hat{\theta}} \tag{5}$$

is the Jacobian of $f(\theta)$ evaluated at the computed estimate $\hat{\theta}$. We therefore get

$$V(\hat{\theta}) = \mathcal{I}(\hat{\theta})^{-1} = (J_{\hat{\theta}}^T \Sigma^{-1} J_{\hat{\theta}})^{-1} \tag{6}$$

Random Effects. To estimate the variance at each location, we first obtain an empirical variance-covariance matrix using the random effect fits by location, denoted by V_0.

Given a location with no observations, its uncertainty will be driven by V_0, which captures the variation across location. However, if a location has data, we can obtain a location-specific fit and uncertainty estimates using the location-specific likelihood. That is, with the prior V_0, the likelihood changes to

$$\hat{\theta}_i = \arg\min_{\theta} := \frac{1}{2}\theta^T V_0^{-1} \theta + \frac{1}{2\sigma^2} \|y_i - f_i(\theta; X_i)\|^2_{\Sigma_i^{-1}}$$

and then we have

$$V_i(\hat{\theta}) = ((J_i)_{\hat{\theta}}^T \Sigma_i^{-1} (J_i)_{\hat{\theta}} + V_0^{-1})^{-1}. \tag{7}$$

9.2. Predictive Validity

The newer approach `CurveFit` uses to estimate uncertainty is based on studying how the model performs in predicting deaths out of sample, and generalizing that performance into the future. The framework is agnostic to the model, that is, any model that generates forecasts can be used. The key invariant is that when obtaining residuals for a specific location, all the other data for all the other locations are available to the model for the estimation. The main goal is to evaluate how well the model predicts for future time points in a location given everything we know so far up to the current time point.

The natural quantities to consider when analyzing and generalizing these errors are

- How many data points we have, and
- How far out we are forecasting.

To obtain the out of sample errors, for each location, we hold out part of the existing data points and compute the residual between the held out data and the fitted curves. We iterate this process, first holding out all data points except the first point, all the way through to only holding out the last data point, fitting on all others [4]. After this analysis, for each location, we have a triangular residual matrix with one axis corresponding to the number of data points used to fit the curve and the other axis represents how far are we predicting out. Using mathematical notation, we have:

$$r^l_{n,i} = \text{pred}^l_{n,t^l_n+i} - \text{obs}^l_{t^l_n+i}, \quad i = 1, \dots \tag{8}$$

where l is the index of location, n is the number of data points, t^l_n is the time index for the n-th data point in location l, and i represents how far we are predicting into the future. Table 2 shows a simple hypothetical example how these residuals would be tabulated across two locations with 5 and 6 datapoints.

Table 2. Tabulating estimation errors at two hypothetical locations with 5 and 6 total datapoints.

Using datapoints:					
5	$\{r^2_{5,1}\}$				
4	$\{r^1_{4,1}, r^2_{4,1}\}$	$\{r^2_{4,2}\}$			
3	$\{r^1_{3,1}, r^2_{3,1}\}$	$\{r^1_{3,2}, r^2_{3,2}\}$	$\{r^2_{3,3}\}$		
2	$\{r^1_{2,1}, r^2_{2,1}\}$	$\{r^1_{2,2}, r^2_{2,2}\}$	$\{r^1_{2,3}, r^2_{2,3}\}$	$\{r^2_{2,4}\}$	
1	$\{r^1_{1,1}, r^2_{1,1}\}$	$\{r^1_{1,2}, r^2_{1,2}\}$	$\{r^1_{1,3}, r^2_{1,3}\}$	$\{r^1_{1,4}, r^2_{1,4}\}$	$\{r^2_{1,5}\}$
Predicting out:	1	2	3	4	5

Prediction space. The evaluation of residuals in (8) can be done in any space, not only to spaces where we fit the data. Specifically, in the current models we fit the data in the log cumulative death rate space, and evaluate the residual in the log daily death rate space. Log cumulative space is more robust to vagaries of the data, but we want to evaluate predictions in log daily death, and we expect less correlated residuals in log daily death space.

Aggregation and smoothing. To account for low data availability for specific locations we choose to analyze residuals in across all locations together rather than in specific locations. More specifically, if one location only has three data points, in order to understand how well we will predict 10 time points into the future past those three data points for this location, we need to utilize information about predictive validity from other locations with more data where we have held out all but the first three data points and predicted 10 time points into the future.

To do this aggregation over location of the residual matrix, for each number of data point n and forecasting horizon i, we obtain mean and standard deviation of the residual by,

$$\mu_{n,i} = \text{mean}(\{r^{\hat{l}}_{\hat{n},\hat{i}} : |\hat{n} - n| \le a, |\hat{i} - i| \le b\})$$
$$\sigma_{n,i} = \text{std}(\{r^{\hat{l}}_{\hat{n},\hat{i}} : |\hat{n} - n| \le a, |\hat{i} - i| \le b\})$$

where a and b are the window size for the number of data points and forecasting horizon, and we include the data across locations when compute the mean and standard deviation. To get the estimates, we use $a = b = 5$. Since some number of data points n and forecasting horizons i pairs only have a couple of contributing locations (for example, only a handful of locations have over 30 data points), we average the mean and standard deviations obtained from the aggregation step over the same window size. After smoothing, we have clearer trends in the relationship between the number of data points, the forecasting horizon and the standard deviation of the residuals. An example of the result of this aggregation and smoothing process is shown in Figure 2.

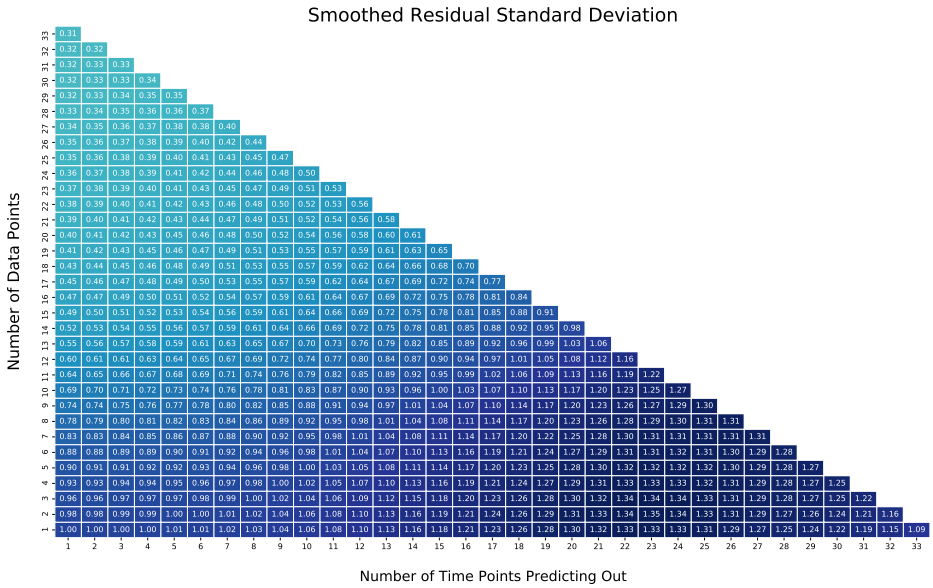

Figure 2. Smoothed standard deviation matrix.

Extrapolating averaged mean, standard deviation, and coefficient of variation values. We need to extrapolate the above matrix to new prediction horizon and number of data point combinations. For example, in Table 2, we don't have any predictive validity results where we had 5 data points and predicted out 5 into the future. In the current approach we use a simple extrapolation technique to extend this table, first extrapolating available quantities to the right, and then down. Continuing with the example in Table 2, we get the array in Table 3.

Table 3. Extrapolating residual matrices to new prediction horizons and number of datapoints

Pred / num:	1	2	3	4	5	6	7	⇒
⇑	⇑	⇑	⇑					
6	$\hat{\sigma}_{6,1} = \sigma_{5,1}$	$\hat{\sigma}_{6,2} = \sigma_{5,1}$	⇒					
5	$\sigma_{5,1}$	$\hat{\sigma}_{5,2} = \sigma_{5,1}$	$\hat{\sigma}_{5,3} = \sigma_{5,1}$	⇒				
4	$\sigma_{4,1}$	$\sigma_{4,2}$	$\hat{\sigma}_{4,3} = \sigma_{4,2}$	$\hat{\sigma}_{4,4} = \sigma_{4,2}$	⇒			
3	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\hat{\sigma}_{3,4} = \sigma_{3,3}$	$\hat{\sigma}_{3,5} = \sigma_{3,3}$	⇒		
2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\hat{\sigma}_{2,5} = \sigma_{2,4}$	$\hat{\sigma}_{2,6} = \sigma_{2,4}$	⇒	
1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\sigma_{1,5}$	$\hat{\sigma}_{1,6} = \sigma_{1,5}$	$\hat{\sigma}_{1,7} = \sigma_{1,5}$	⇒

Generating draws for predictive validity-based uncertainty. Once we have residual standard deviation computed across all observed values of forecast horizon and number of data points, and extrapolated to future values, we generate random errors appropriately around the mean curve to simulate draws.

Specifically, for one draw, we generate one realization from a standard normal distribution and then add on that amount of noise scaled by the standard deviation from Table 3 to the mean curve for each prediction horizon, given the amount of data currently observed for that location. We do this for any number of draws (for a given model this will typically be ≥ 200 draws). Currently, we are only incorporating standard deviation of the residuals into the uncertainty and not the mean of the residuals.

10. Peak Analysis

In this section, we describe analyses to detect which locations have peaked, and what the likely durations of these peaks might be. The technology to do this uses splines, and a brief primer on splines is first provided in Section 10.1. The peak detector is then briefly described in Section 10.2, while the duration detector is described in Section 10.3.

10.1. Splines and Spline Shape Constraints

A spline basis is a set of piecewise polynomial functions with designated degree and domain. If we denote polynomial order by p, and the number of knots by k, we need $p + k$ basis elements s_j^p, which can be generated recursively.

Given such a basis, we can represent any dose-response relationship as the linear combination of the spline basis elements, with coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p+k}$ that are fit to data:

$$f(t) = \sum_{j=1}^{p+k} \beta_j^p s_j^p(t). \tag{9}$$

We can impose shape constraints such as monotonicity, concavity, and convexity on splines. Constraints on splines have been developed in the past through reformulation techniques, see e.g. [8]. We use explicit constraints instead.

Monotonicity. Spline monotonicity across the domain of interest follows from monotonicity of the spline coefficients [1]. Given coefficients

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix},$$

the curve $f(t)$ in (9) is monotonically nondecreasing when

$$\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$$

and *monotonically non-increasing* if

$$\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n.$$

Convexity and Concavity. For any twice continuously differentiable function $f : \mathbb{R} \to \mathbb{R}$, convexity and concavity are captured by the signs of the second derivative. Specifically, f is convex if $f''(t) \geq 0$ is everywhere, an concave if $f''(t) \leq 0$ everywhere. We can compute $f''(t)$ for each interval, and impose linear inequality constraints on these expressions.

10.2. Peak Detector

When running the model, we use peaked locations to obtain relationships between peaks and social distancing covariates. Here we detail an automatic peak detector to give a list of potential peaked locations for further expert vetting. For example, from the data set from 04/10/2020, the detector selects 31 candidates from 107 locations, largely reduced the search space, and then expert consensus is used to select the final 19 locations from this reduced set.

The detector works as follows. Since the cumulative death rate is modeled using the ERF function, we know that the log daily death rate should roughly follow a quadratic function with negative curvature. When the location reaches its peak, the log daily death curve should have either almost reached or passed the part of this curve where the tangent line is horizontal, see e.g. Emilia-Romagna in Figure 3.

To detect whether this has happened, we fit a quadratic B-spline to each location in the log daily death rate space using the `Xspline` package [9], and compute the minimum of the absolute value of the derivatives of the fitted curve. We use two knots, at 0 and 100; `Xspline` allows functionality for computing derivatives of any fitted splines. To detect whether a location has peaked, we choose a threshold and declare peaks when the minimum absolute derivative is less than this threshold (we use `5e-3` to get 31 locations). To make the detector more accurate, we impose the requirement that the second order derivative of the spline should be negative and we require the number of the observations has to be greater or equal to 20.

10.3. Peak Duration

As more and more locations starting to decline in the daily death, we observe that many locations have a flat peak of variable duration. To estimate the duration of the peak, we extend the idea of the peak detector, fitting a concave quadratic spline in the log daily death space, also using the `Xspline` package. This approach can capture the flat shape at the top of the peak, while denoising the data through the concavity assumption. After fitting the spline, we compute derivatives of the curve in the log daily death space. Given a threshold, we obtain the duration of the peak by the difference between points where the relative derivative (as a fraction of maximum observed derivative) crosses the threshold on each side of the peak.

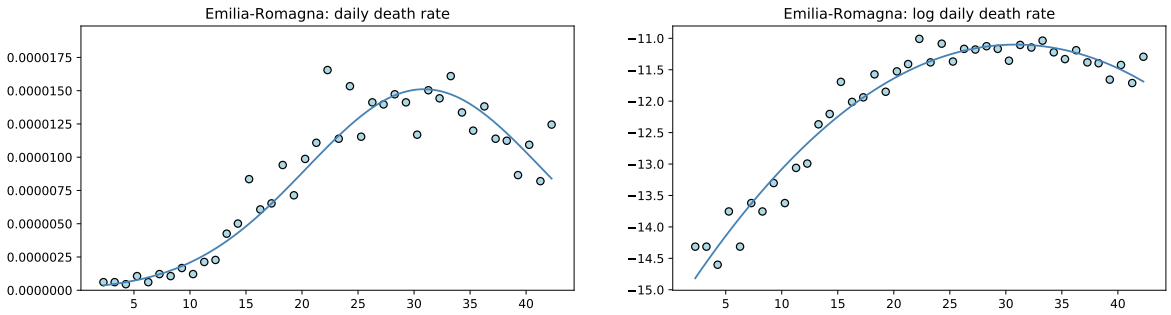

Figure 3. Peaked detector example for Emilia-Romagana. Dots are the data in daily and log daily death space and curve are the spline fits.

11. Model Specification for Estimates

The final results use an a model ensemble, where models in the ensemble differ by definition of the social distancing covariate. Final estimates and uncertainty are created at the draw level. When we have fewer than 18 datapoints, each draw from a particular model (using a particular social distancing covariate) interpolates draws between short-range and long-range models. When we have 18 or more datapoints, we use the linear Gaussian extended model. These analyses are explained in detail below, along with common settings and assumptions. At the end we document the ensemble.

11.1. Data processed outside of the JHU Pipeline

France. Due to out-of-hospital deaths being reported differentially to in-facility deaths in France, we have been redistributing French data. Using data from Sante Publique[7] cumulative deaths in hospital are kept distinct from deaths reported in EHPAD (Établissement d'hébergement pour personnes âgées dépendantes) and EMS (Établissements medico-sociaux). We have redistributed the deaths reported in the latter sector proportionate to the daily deaths reported in hospitals.

Spanish subnationals. With subnational locations in Spain missing from JHU, we have instead used the Daily governmental reports from the Centro de Coordinación de Aleras y Emergencias Sanitarias (CCASES)[8].

Catalonia addendum. In the Spanish governmental report Number 78 dated 17th April 2020, it was noted that there was a discrepancy between reported tabulations, and that reported by Salud Pública de Cataluña (Sub-direcció General de Vigilància I Reposta a Emergències de Salut Pública). For the epidemiological dates 16th April onwards, we instead report the number of deaths indicated by the Catalonian Government instead[9]

Germany. With subnational locations in Germany missing from JHU, we have instead used the daily epidemiological reports from the Robert Koch Institute[10]

Wuhan City, Hubei Province, China. With sub-provincial data missing from JHU, we have instead manually extracted the time series of deaths as reported by the Health Commission of Hubei Province[11] in their daily situation report press releases.

Wuhan City addendum. On the 16th April 2020, Wuhan City death numbers were increased by 1290 deaths, and cases by 325. We have subtracted these numbers from the subsequent days of reported cases and deaths since these deaths are known not to have occurred on the 16th April 2020, but across the months previously. We are currently withholding these deaths from the model.

United States.

- Illinois. Due to repeated inconsistences in reported cumulative total deaths between JHU and the Illinois Department of Public Health, we replaced the JHU time series with one derived from the Illinois Department of Public Health instead[12]. Given the lack of an historical archive,

[7] https://dashboard.covid19.data.gouv.fr/
[8] https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov-China/situacionActual.htm
[9] https://analisi.transparenciacatalunya.cat/Salut/Incid-ncia-de-la-COVID-19-a-Catalunya/623z-r97q
[10] https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Gesamt.html
[11] http://wjw.hubei.gov.cn/fbjd/dtyw/
[12] http://www.dph.illinois.gov/topics-services/diseases-and-conditions/diseases-a-z-list/coronavirus

we used The COVID Tracking Project's[13] archive of preserved screenshots to reconstruct the historical time series of reported total cumulative deaths.

- New York. Due to the mid-outbreak of stratification of confirmed and probable deaths in New York City, we derived an alternative data processing workflow for New York City and therefore New York State. We replaced the JHU New York City time series with the New York Times New York City time series[14] which more closely tracks with the time series of confirmed deaths as indicated by New York City Health[15]. To account for the reporting of probable deaths, for the most recent day of reporting, we take the difference between the New York City Health total number of deaths (i.e. the sum of probable and confirmed deaths) and subtract the New York Times reported deaths for that day, and re-distribute the remainder proportionate to the daily deaths reported by New York Times.
- Washington. Due to the unique high-intensity epidemic in the Life Care Kirkland facility in Washington state [6, 3] we have modeled this facility separately from the general population. Furthermore, as our initial development of the model was focused on King and Snohomish counties in Washington state, we have also stratified these 2 counties from the rest of Washington state. In other words, for Washington state, we model 3 populations explicitly: (i) the Life Care Kirkland facility; (ii) the remainder of the King and Snohomish county population; and (iii) all other counties in Washington state. Data was collected directly from each County Health Department, with metadata on whether deaths were reported from the Life Care Kirkland facility retained.

11.2. Pre-processing

11.2.1. Short-term Pseudo-Death data from Hospitalizations

We use what we know about the timing of the disease to generate additional short-term predicted deaths (pseudo-data) from hospitalizations and use these in our model. On average, the time between hospitalization and death is 8 days. Using location-specific hospitalization data which has more than 10 deaths, we build simple measure that can help predict deaths:

$$R_{d/h} = \frac{\text{cumulative deaths up to time } t}{\text{cumulative cases up to time } t-8}$$

If a location has more than 10 deaths, we then use a location-specific ratio and current case loads to generate 'pseudo-data' for the next 8 days, and incorporate this pseudo-data into the model, with a fractional weight of $\frac{1}{5}$ so the model fits to real data much more strongly than pseudo-data. If a location has fewer than 10 days, we use the average ratio and location-specific cases to predict location-specific deaths in the next 8 days.

11.2.2. Moving average smoothing of daily deaths

We use a 3-day moving average across times $(t-1, t, t+1)$ in the space where we fit the model, log age-standardized cumulative death rate. For the first day, where $t = 0$, we project the average difference in smoothed values from $t = 1$to $t = 3$ back from $t = 1$. For the last day, where $t = N$, we project the average difference in smoothed values from $t = N-3$ to $t = N-1$ forward. We drop the last day from analysis if there are no new deaths reported.

11.3. Model functional form, variables, and bounds.

All of the models in the ensemble follow (3).

Measurement model.

$$\log(\text{cumulative death rate in location } l \text{ at time } t) = \log(D(t; \alpha^l, \beta^l, p^l)) + \epsilon_t^l$$

Statistical model for parameters.

$$\begin{aligned}
\alpha^l &= \exp(\mu_\alpha + u_\alpha^l) \\
\beta_l &= (\mu_\gamma + u_\gamma^l)\text{Covariate}^l \\
p^l &= \exp(\mu_p + u_p^l)
\end{aligned}$$

[13] https://covidtracking.com/
[14] https://github.com/nytimes/covid-19-data
[15] https://www1.nyc.gov/site/doh/covid/covid-19-data.page

Error model.

$$\begin{aligned} \epsilon_t^l &\sim N(0, \sigma_t^2) \\ u_\alpha^l &\sim N(0, \sigma_\alpha^2) \\ u_\gamma^l &\sim N(0, \sigma_\gamma^2) \\ u_p^l &\sim N(0, \sigma_p^2). \end{aligned} \tag{10}$$

Simple bound constraints are also used, and Table 4 shows bounds that apply to all models. The error model assumptions are set differentially, depending on the model, as explained in the next sections.

Table 4. Parameter bounds and prior values common across all models. * The interpretation for β assumes the same value of the social distancing covariate value as in Wuhan (normalized to 1).

Parameter	Bounds	Interpretation
μ_α	$(-\infty, 0]$	$0 \leq \alpha \leq 1$
μ_γ	$[15, 100]$	$15 \leq \beta \leq 100^\star$
μ_p	$[-15, -6]$	$\exp(-15) \leq p \leq \exp(-6)$

11.4. Low-Data Case: Fewer Than 18 Daily Death Datapoints

For locations that have fewer than 18 points of daily data, we generate forecasts that transition from short-term to long-term models. This also is the way all forecasts were generated before the April 17, 2020 update, so we give full details below.

11.4.1. Short-term models

Short-term models are specified to fit the data. In order to obtain location specific models we

- First fit peaked locations jointly to get a prior distribution
- Fit to individual locations using the prior we obtained from peaked locations.

Fitting to peaked locations. In order to obtain some of the statistics (10), we first fit a joint model on the 'peaked' locations, obtained using the peak detector in Section 10.2 followed by expert vetting of the candidates. To consider later points more than earlier points, we set

$$\sigma_t = \frac{1}{0.1 + t^2}. \tag{11}$$

With this specification of measurement error, we fit the joint model with bounds from Table 4 and set $\sigma_\alpha = \sigma_p = \infty$, and $\sigma_\gamma = 10$. From the resulting empirical distribution of γ_l in the peaked locations, we then get a mean $\overline{\mu}_\gamma$ and standard deviation σ_γ that we can use as a prior when fitting individual locations.

Fitting individual locations. The individual fits are done completely independently, so each location is fit with its own fixed-effects only model:

$$\begin{aligned} \alpha^l &= \exp(\mu_\alpha^l) \\ \beta_l &= (\mu_\gamma^l)\text{Covariate}^l \\ p^l &= \exp(\mu_p^l) \\ \mu_\gamma^l &\sim N(\overline{\mu}_\gamma, \sigma_\gamma^2) \end{aligned}$$

with only the prior on μ_γ^l informed by the joint fit. The variables (α^l, p^l) can adapt to each location, still subject to bounds in Table 4. The standard deviations are still given by (11).

11.4.2. Long-term models

The purpose of long-term models is to forecast far away, following more closely those locations that have already peaked. Just as in the short-term case, the strategy is

- First fit peaked locations jointly to get a prior distribution
- Fit to individual locations using the prior we obtained from peaked locations.

The list of peaked locations is the same as for the short-term models, but the remaining specifications are different.

Fitting to peaked locations. In order to obtain some of the statistics (10), we again fit a joint model on the 'peaked' locations.

For long-term models, we let standard errors follow a different functional form, that still emphasizes the latter points but not as strongly:

$$\sigma_t = \frac{1}{1.0 + t} \tag{12}$$

We also let the strength of the σ_γ depend on the timeliness of the datapoint, so later values have more influence on the inferred multipliers. Specifically we use the formula

$$\sigma_\gamma(t) = 10^{\min(0,\max(-1,t/10-1.5))}, \tag{13}$$

which varies between 0.1 and 1, in contrast to the value 10 used in the short-term model.

Finally, for the tight model we use a functional prior (see Section 6.3)

$$\alpha\beta \sim N(\exp(0.7), 0.1) \tag{14}$$

where the value $\exp(0.7)$ was obtained by fitting a regression in log-space for the quantity $\alpha\beta$ to the slopes at $t = 14$ days for data rich locations. We impose a prior on $\alpha\beta$ because this term determines the behavior of slopes of the trajectory of daily deahts $D''(t)$, see e.g. (2).

The peaked locations again determine a mean $\overline{\mu}_\gamma$ and standard deviation σ_γ that we use as a prior when fitting individual locations.

Fitting to individual locations. The individual fits are again done independently, so each location is fit with its own fixed-effects only model:

$$\begin{aligned}
\alpha^l &= \exp(\mu_\alpha^l) \\
\beta_l &= (\mu_\gamma^l)\text{Covariate}^l \\
p^l &= \exp(\mu_p^l) \\
\mu_\gamma^l &\sim N(\overline{\mu}_\gamma, \sigma_\gamma^2)
\end{aligned}$$

with only the prior on μ_γ^l informed by the joint fit (using the long-term specifications). The variables (α^l, p^l) can adapt to each location, still subject to bounds in Table 4. The standard deviations are given by (12), and the functional prior (14) is also used for each individual location.

11.4.3. Combining draws from long-term and short-term models

For each location, the previous sections explain how we get long-term and shor-term location-specific fitted models, that are informed by priors estimated using peaked locations. Given a location, we use the predicted validity framework of Section 9.2 to obtain 200 draws from each of the long-term and short-term location-specific variants.

To create the combined 200 draws that transition smoothly from the short-term to the long-term regime, we use simple linear interpolation in log increment space:

$$\text{increment of log D} = \lambda(t)[\text{increment of log D (long)}] + (1 - \lambda(t))[\text{increment of log D (short)}]$$

where

$$\lambda(t) = \min\left(1, \max\left(0, \frac{t - t_s}{t_e - t_s}\right)\right).$$

and where t_s and t_e are start and end times for the period of interest, starting with the last datapoint and continuing to the end of the forecast horizon. The resulting draw is then constructed by aggregating the joint increments over (t_s, t_e).

We illustrate these steps all together using New York as an example. Figure 4 shows four plots. The short-term models are everywhere indicated using a red curve, while the long-term models are shown using green. The blue curve interpolates between these at the draw level in daily death space. Uncertainty in the plots is generated using the predictive validity framework, as described in Sections 6 and 7.

11.5. Default Case: 18 or More Daily Death Datapoints

For all locations where we have 18 or more datapoints, we no longer use the short-term strategy. Instead we use the extended model strategy detailed in Section 8.

Specifically, we follow the following steps:

- Fit a long-term specification as described in Section 11.4.2. For each location, this gives a Gaussian atom that has its own (γ, α, p) parameters.

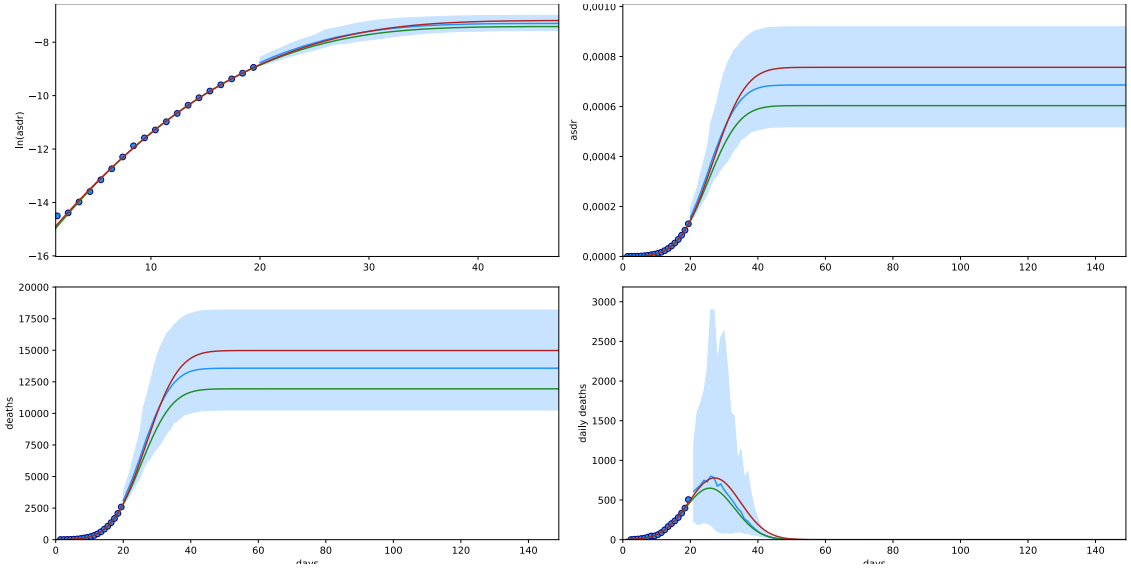

Figure 4. New York fits using the strategy in Section 11.4 (analysis and data from April 6). Top left: log cumulative death rate. Top right: cumulative death rate. Bottom right: cumulative deaths. Bottom left: daily deaths. Uncertainty using the PV framework is shown using blue shading. The short-term model is indicated by the red curve, while the long-term model is indicated by the green curve. The mean forecast, shown using the blue line, interpolates between the short-term and long-term models in daily death space.

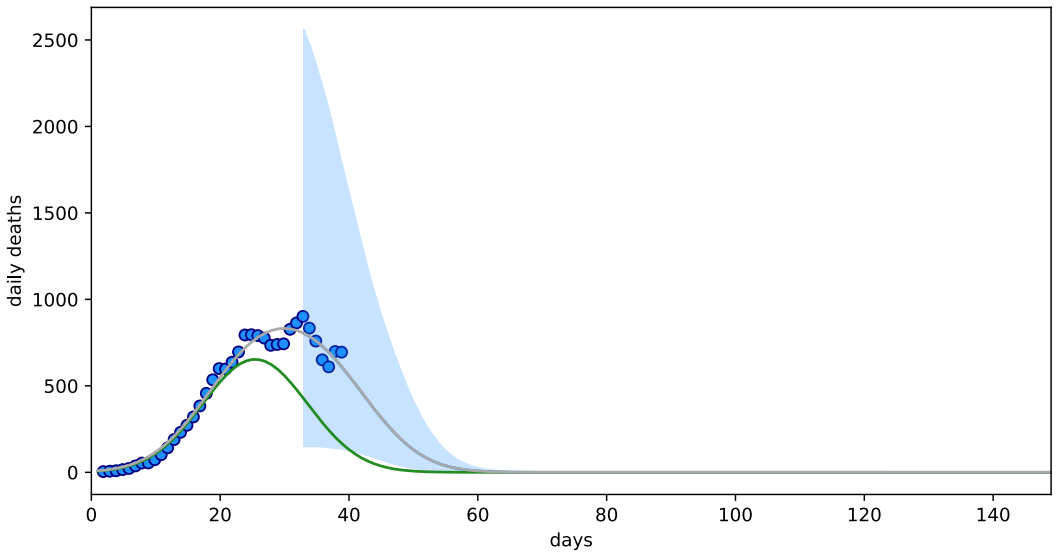

Figure 5. NY fits in daily death space, using the strategy in Section 11.5 (analysis and data from April 16). The long-term model, shown in green, is strongly tied to the social distancing covariate under-estimates the deaths time series and cannot adjust to the peak duration. The fitted linear combination of Gaussians, shown in grey, is fit as described in Section 8, uses the green fit as an atom, and fits much better to the data. Uncertainty estimates (shown using blue shading) for the entire procedure are obtained through the predictive validity framework described in Section 9.

- Fit location-specific combination of Gaussians using the 13 staggered peaks strategy given in Section 8.

The effects of this approach are as follows:

1. We borrow strength across locations in obtaining the relationship between the social distancing covariate and the peak times for places that have peaked.
2. We obtain location-specific Gaussian atoms that use the borrowed strength from the first step, and adjust the shape of the Gaussian atom to each specific location.
3. The final location-specific forecasts for data-rich locations use a combination of these atoms fit to the data at each location, captures individual variation, including asymmetric epidemic shapes, flat peaks, and other anomalies.

11.6. Ensemble over different covariate definitions.

The final estimates are created by an ensemble, at the draw level, across different model types. Models differed by definition of the social distancing covariate. The construction of these covariates (both for the initial and more recent estimates) is described in Section 5.

Once we have a set of covariates to ensemble over, the statistical specification and fitting procedure of each model type is specified exactly as in the previous sections. The final ensemble was created by equally weighting draws from each type of covariate model. The process is illustrated for New York in Figure 6 showing differences in data, analysis, and covariates between April 6th and April 16th.

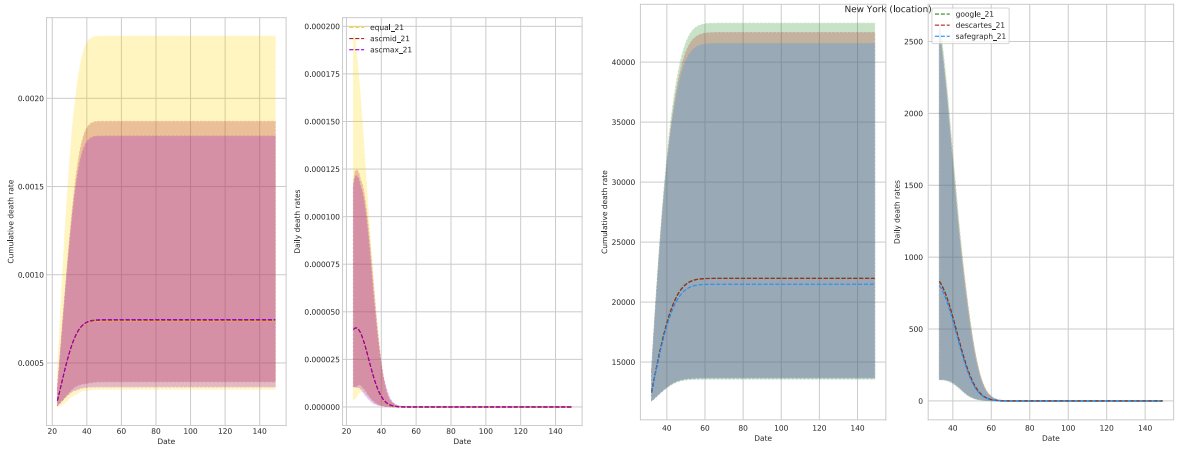

Figure 6. New York forecasts of three covariate models that are incorporated into a final ensemble. Left panel: data, analysis, and covariates from April 6th for cumulative and daily death rates, using the analysis detailed in Section 11.4. Right panel: data, analysis, and covariates from April 16 for cumulative and daily death counts using the analysis detailed in Section 11.5. Covariate definitions for these dates are described in Section 5.

References

[1] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[2] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*, volume 105. Siam, 2008.

[3] J. Healy and S. Kovaleski. The coronavirus's rampage through a suburban nursing home. *N.Y. Times, https://www.nytimes.com/2020/03/21/us/coronavirus-nursing-home-kirkland-life-care.html, Accessed on 03/22/2020*, 2020.

[4] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts: Melbourne, Australia, 2018.

[5] J. R. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3):245–262, 2003.

[6] T. M. McMichael. Covid-19 in a long-term care facility—king county, washington, february 27–march 9, 2020. *MMWR. Morbidity and Mortality Weekly Report*, 69, 2020.

[7] J. A. Nelder and R. W. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[8] N. Pya and S. N. Wood. Shape constrained additive models. *Statistics and Computing*, 25(3):543–559, 2015.

[9] P. Zheng. Xspline package, https://github.com/zhengp0/xspline, 2020.

[10] P. Zheng, A. Y. Aravkin, R. Barber, R. J. Sorensen, and C. J. Murray. Trimmed constrained mixed effects models: Formulations and algorithms. *arXiv preprint arXiv:1909.10700*, 2019.

[11] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.