
Real-Time Backfill Correction of COVID-19 Infections

Anonymous

Abstract

1 Introduction

As a global public health issue resulting in extreme societal, economic, and political disruption, the COVID-19 pandemic has highlighted the importance of proactive epidemic forecasting, critical in providing response guidance for policymakers and healthcare authorities. However, the latency problem commonly exists in public health surveillance data streams. It often causes imperfect or even error-prone situational awareness [1][2] and brings challenges to real-time forecasting accuracy.

To be more detailed, the latency problem refers to the situation where the observations or statistics provided by the surveillance systems are most up-to-date but not finalized. The initially released data is usually revised/corrected through a process known as the *backfill* phenomenon. Except for errata, some data will only be reported once with some degree of reporting delay, which is regarded as a special case of backfill. The term *backfill correction* we used throughout this paper refers to providing projection of finalized values based on the current observations.

Many factors can affect the magnitude of backfill, such as laboratory confirmation, infrastructure difficulties, coordination efficiency between health authorities and government[3][4], etc. Understanding the backfill pattern is crucial, as it could lead to intelligent strategies for making accurate projection of the observations that are far from finalized. Backfill has been commonly mentioned in public health studies, ranging from influenza[5], dengue[6] to COVID-19 forecasting[7]. Different approaches have been proposed to address the backfill problem retrospectively and in real time. One common strategy is to re-scale real-time reports by estimating scaling factors from historical data[8][9]. Other ones such as backcasting handle the reporting error by generating past weeks' validation data using the most recent data reports for those weeks and accounting for uncertainty in reporting patterns[10]. Nowcasting studies deal with backfill via complicated statistical or machine learning methods[11][12] strictly based on incomplete data.

It is worth noting that backfill correction is more challenging for COVID-19 from two perspectives: 1) SARS-CoV-2 is constantly changing, and the dominant variant is continuously updated while the epidemic pattern of seasonal influenza is more stable and less noisy; 2) the degree of concern and response measures from the government and healthcare authorities to COVID-19 is constantly changing as well, making the pattern of data collation and revision difficult to predict. Despite this, digital surveillance, developed in the recent years especially during the COVID-19 pandemic, play an indispensable role in understanding and modeling backfill patterns. Data revision history is no longer ignored but well captured and archived with support of new technology and methods.

In this paper, we focus on making projection of the finalized values of observations in *real-time* and measuring the uncertainty in projection, laying out a framework for an operational backfill correction system using provisional data. At any given date s , we are forced to use data that is guaranteed to be available at date s . Specifically, backfill correction takes as input all the revisions of observations with finalized values included as the target, and output the estimation of probability distribution

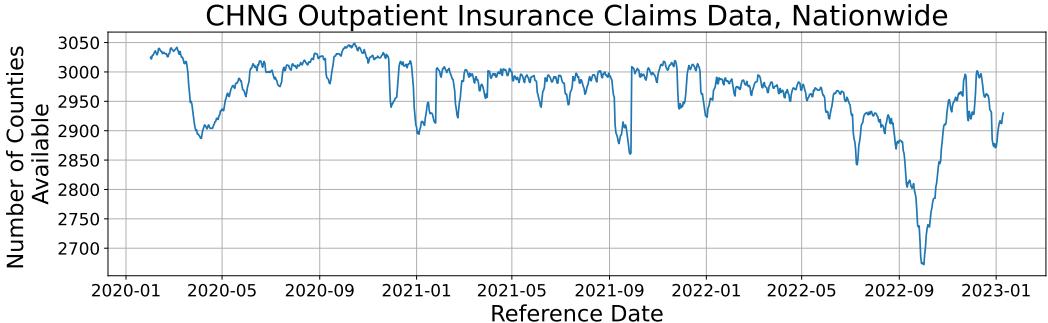


Figure 1: The number of unique counties available in U.S. over time in CHNG’s de-identified COVID-19 claims data

over the target variable based on the most up-to-date observations. There are two categories of observations that we mainly care about: 1) counts; 2) fractions. Take COVID-19 insurance claims data as an example, counts can be the number of outpatient insurance claims that’s made for COVID-19 patients while fractions can be the percentage of COVID outpatient insurance claims in terms of total outpatient insurance claims.

An outline for this paper is as follows. In Section 2, we provide preliminary information about notations and the problem setup. The details of the model are described in Section 3. Section 4 covers evaluation of the results and comparing the backfill correction results made in real-time to those without correction. In Section 5, we conclude with a discussion and describe a few directions for future work.

2 Preliminaries

Given y_{it} an epidemic observation that have onset at time t for location i , $y_{i,1:t}$ is the time series observations of the uni-variate surveillance. Due to the existence of backfill, each y_{it} has its *backfill sequence*[13] $y_{it}^{t:s}$ as of date s ($s \geq t$). To be clear on nomenclature, throughout this paper, we use the term *reference date* for date t referring to the date when the observations have onset and the term *report date* for date s referring to the date when the observations are reported.

For convenience, we introduce another term lag l defined as $l = s - t \in \mathbb{N}$, which is the number of days between the report date s and the reference date t . Now we change our notation henceforth and rewrite y_{it}^s as y_{itl} . y_{itl} is also named to be the $(l + 1)$ th release of y_{it} . The backfill sequence $y_{it}^{t:s}$ is equivalent to $y_{it,0:l}$.

Only a tiny part of data revisions occurs when data is lost but found later or when data is entered incorrectly but changed later. Most of them are caused by reporting delays. Therefore, the backfill sequence $y_{it,0:l}$ is guaranteed to converge to a particular value Y_{it} when $l \geq L_{it}$, L_{it} is sufficiently large. Y_{it} is called the finalized value of y_{it} .

In this paper, we mainly use Change HealthCare (CHNG) ’s insurance claims data which is available for more than two thousand most populous US counties, accounting for more than 45% of the U.S. population in general. Multiple pre-processed signals based on this data are publicly available through Delphi’s COVIDcast API [14]. In order to construct the raw backfill data (including all historical versions), we start from the restricted line-level de-identified outpatient insurance claims data. It is worth pointing out that the data has geographic details down to 5-digit ZipCode level and is updated on a daily basis with no latency. We assume that a missing report is the same as reporting 0.

A useful way we propose to look at backfill pattern is by examining the variable p_{itl} defined as

$$p_{itl} = \frac{y_{itl}}{Y_{it}} \times 100\%$$

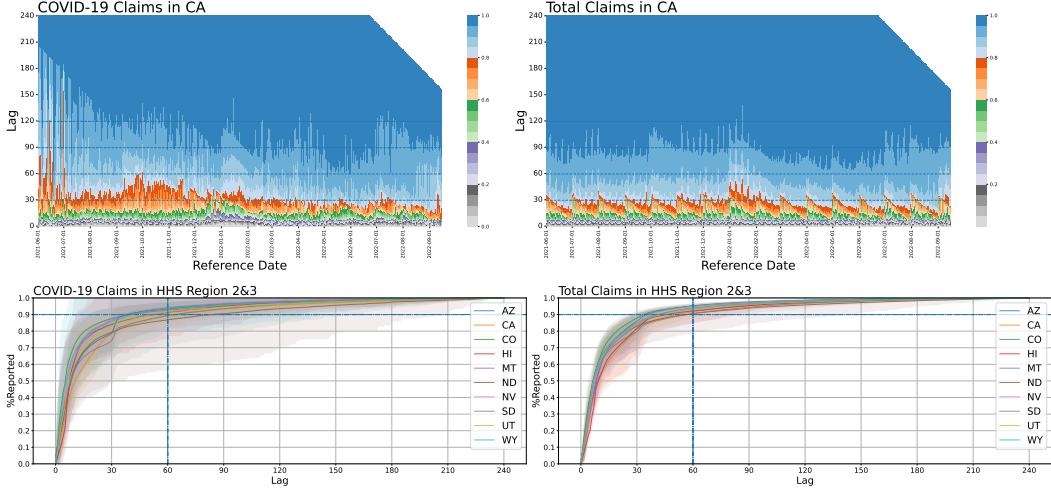


Figure 2: Top: percentage of reported distribution in New York, grouped by lag, and recorded over reference dates. Bottom: Mean percentage of reported, averaged over all reference dates between 2021-04-01 and 2023-01-10, over lags for states in HHS Region 2&3. The shaded bands correspond to 5% quantile to 95% quantile intervals. The left panel in the figure is based on CHNG outpatient COVID-19 claims data while the right panel is based on CHNG outpatient total claims data.

p_{itl} describes the percentage of claims that have been reported after the $(l + 1)$ th release for location i and reference date t . Since Y_{it} is a theoretical value which is not obtainable directly, we use y_{it}^s ($s = 2023-03-11$) to approximate it.

The distribution of p_{itl} exhibiting how the backfill sequences vary over time. The day-of-week effect and week-of-month effect are evident, since the revision trends in lags for a fixed level of %reported show spikes and bumps upward at the end of each week and around the beginning of each month. The reporting delay is prominent during the time period when the epidemic curve is at its peak. The top panel in Figure X displays results to this end using data in New York as an example. The backfill pattern is much more stable over time for total claims compared to COVID-19 claims. Overall, COVID-19 reporting is less efficient than Non-COVID-19 reporting. The variance is larger in the backfill pattern of COVID-19 claims which increases the difficulty in backfill correction.

Furthermore, there is a high degree of heterogeneity in the backfill pattern across locations. For most of the locations (states and populous counties), the data revision tends to occur in the first two weeks of the backfill sequences to a large extent. The bottom panel in the figure shows that nearly all the mean %reported of COVID-19 reaches 90% when lag equals to 60 for states in HHS Region 2 and Region 3 except New Jersey.

2.1 Defining the Target

Theoretically, the limit values of the backfill sequences are the target to be projected. However, the practice suggests that L_{it} is highly inconsistent for different reference dates and can be incredibly large. Figure X gives an illustration where most of the states require more than 180 days (around half a year) for the backfill sequences of the COVID-19 claims reports with reference date=2021-08-01 to converge. If we set the projection target to be Y_{it} , we are forced to use the backfill sequences of y_{it}^s with $t \leq s - L_{it}$ only for the training data on each testing date s . This means the backfill pattern captured based on the empirical distribution of all available data is very likely to be biased against the most recent backfill pattern, as discussed in the previous subsection. Note that, though L_{it} is far from uniform at random, the distribution of %reported indicates that substantial effective revisions are usually made within the first two months. Thus, we step back, setting a target lag L and using

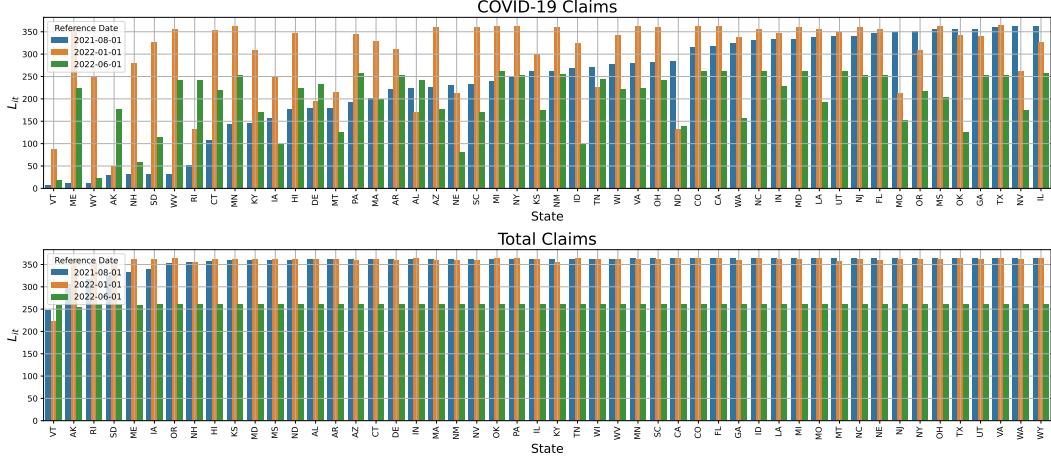


Figure 3: Top: comparison of the smallest lags required for convergence across states, displayed for a sample of different reference dates, based on CHNG outpatient COVID-19 claims data. Bottom: as in the figure on the top, but based on CHNG outpatient total claims data.

Y_{itL} as the projection target. For any location i and reference date t , we assume that

$$\mathbf{E}\left[\frac{Y_{itL} - Y_{it}}{Y_{it}}\right] \leq \epsilon$$

The larger the target lag L is, the smaller ϵ can be. In short, the target lag L selection faces a trade-off between timeliness and accuracy. Out of practical considerations and understanding of CHNG outpatient data, we set $L = 60$ henceforth in our real-time backfill correction experiments.

2.2 Problem Setup

Estimation Period. For every report day t in between 2021-08-30 and 2023-03-11 inclusive (589 days in total), we make projections of the indicator values of reference dates $t, t-1, t-2, \dots, t-(L-1)$ respectively based only on the data that would have been available as of time t .

Geographic Scope. We produce quantile projections for the targets of each reference date at state resolution and additionally produce the estimates for the top 200 populous U.S. counties.

Evaluation. We evaluate all the quantile projections in between 2021-06-01 and 2023-01-10 inclusive (589 days in total) and at each of the locations (50 states and 200 counties) against the target values as defined in section 2.1.

3 Methods

In this section, we propose a non-parametric model based on quantile regression to model backfill patterns and introduce the covariates that help capture the backfill dynamics across time.

3.1 Real-Time Quantile Regression

For any test date S , we are trying to make distributional projection of Y_{itL} for all $t \in [S-L, S]$. We use the notation $Q_{Y_{itL}|X_{itl}}^\tau$ to represent the τ th quantile of Y_{itL} given X_{itl} , where X_{itl} is the vector of features based on $\{y_{itl}\}_{t+l \leq s}$.

$$Q_{Y_{itL}|X_{itl}}^\tau = \inf\{y : F_{Y_{itL}|X_{itl}}(y) \geq \tau\}$$

Motivated by reducing the relative error between the projections and the target, we do a log transformation (natural logarithm) $f(x) = \log(x + a)$ of all the variables. a is a small enough constant to

avoid invalid values. Since $f(\cdot)$ is monotonically increasing,

$$Q_{f(Y_{itL})|X_{itl}}^\tau = f(Q_{Y_{itL}|X_{itl}}^\tau)$$

We now make the assumption that the τ th conditional quantile on log scale is given as a linear function of explanatory variables.

$$Q_{f(Y_{itL})|f(X_{itl})}^\tau = \beta^{\tau T} X_{itl}$$

The covariates included in X_{itl} are the following:

- $\mathbb{1}(\text{wd}(t+l) == k)$: indicator functions checking the day-of-week of the report date $t+l$, where $k \in \{0, 1, 2, \dots, 5\}$.
- $\mathbb{1}(\text{wd}(t) == k)$: indicator functions checking the day-of-week of the report date t , where $k \in \{0, 1, 2, \dots, 5\}$.
- $\mathbb{1}(\text{mw}(t+l) == k)$: indicator functions checking the week-of-month of report date $t+l$, where $k \in \{0, 1, 2, \dots, 3\}$
- $f(\tilde{y}_{itl})$: average of 7-day past values on log scale.
- $\mathbb{1}(\sqrt{y_{itl}} \in \text{interval}_k)$ where interval_k are uniform bins at square-root level
- 7day-slope: $f(\tilde{y}_{itl}) - f(\tilde{y}_{i(t-7)(l+7)})$: the 7-day slope, where we compare the difference between the 7-day average of current observations and the 7-day average of the observation assigned to 7 days ago on log scale.
- 7day-Modification where we compare the difference between the 7-day average of the current raw counts and the 7-day average from the first release.

$$\text{7day-Mod} = f(\tilde{y}_{itl}) - f(\tilde{y}_{ito})$$

- 7day-Prev-Mod-start where we compare the difference between the 7-day average of the raw counts reported a week ago with the same lag and the 7-day average of the raw counts reported a week ago from the first release.

$$\text{7day-Prev-Mod-Start} = f(\tilde{y}_{i(t-7)l}) - f(\tilde{y}_{i(t-7)0})$$

- 7day-Prev-Mod-follow

$$\text{7day-Prev-Mod-Follow} = f(\tilde{y}_{i(t-7)l}) - f(\tilde{y}_{i(t-7)(l+7)})$$

$\text{wd}(\cdot)$ is a function returning the day of the week as integers, where Monday is 0 and Sunday is 6. $\text{mw}(\cdot)$ is a function checking the week of a month. For example, $\text{mw}(\cdot) = 0$ represents the first week of a month that starts from a Sunday. $\tilde{y}_{itl} = \sum_{v=1}^7 y_{i(t-v)(l+v)}$ is the average of 7-day past values reported at the same date $t+l$

We achieve the estimates of coefficients by solving the problem

$$\beta^\tau = \arg \min_{\beta} \sum_{l \in \mathcal{L}, t+l < S} (\rho_\tau(f(Y_{itL}) - X_{itl}\beta)) + \lambda \|\beta\|_1$$

where ρ is the quantile loss[15], $\mathcal{L} = \{l : 0 \leq l \leq L\}$. $\|\cdot\|_1$ is the $_1$ norm. The hyper-parameter $\lambda \geq 0$ controls the complexity of the model.

The model is trained adaptively. At each test date S , a series of new observations is reported which gives a series of y_{itl} s,

$$y_i^S = \underbrace{\{y_{i,S,0}, y_{i,(S-1),1}, y_{i,(S-2),2}, \dots, y_{i,(S-L+1),(L-1)}\}}_{\text{Testing Data}}, \underbrace{\{y_{i,(S-L),L}, \dots\}}_{\text{Target}}$$

. Note that, the backfill sequence of $y_{i,(S-L)}$ is moved from testing to training since the target $y_{i,(S-L),L}$ becomes available.

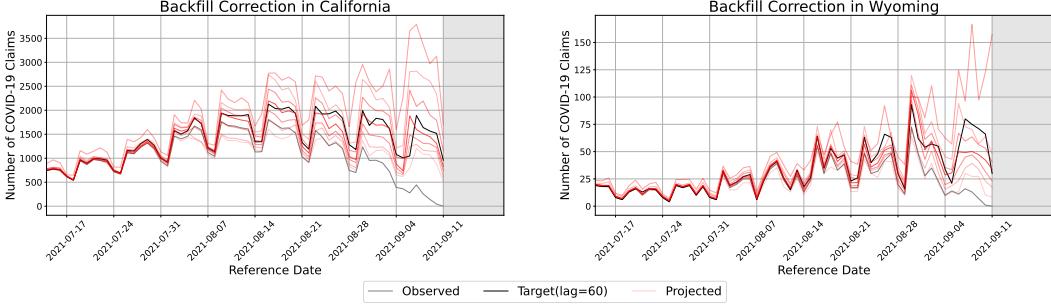


Figure 4: Illustration of estimating target number of CHNG outpatient COVID-19 claims as of 2021-09-11. The gray curve shows the most recent release of the number of COVID claims based on CHNG outpatient data in California and Wyoming. The red lines show the projected quantiles of the 60th revision of the number of COVID-19 claims for each reference date. The black curve shows the actual 60th revision for the values of each reference date in the future.

Shortening the Lag Window As the backfill patterns vary largely over lags, we investigate shortening the lag window in the regularized backfill correction problem and train separate models for observations with different levels of lag. Theoretically, training separate models is the same as training one model with all data pooled together but incorporating indicators to inform the level of lag (The only difference is that, fitting separately generates different variance estimates) in linear regression. However, it is not the case when lasso penalty is added for reducing the complexity of the model and rolling out extreme outliers. When making projection of $y_{itl'}$ (l' is small enough, e.g. $l' \leq 7$), instead of using all the data as described above, we now change the problem to be

$$\beta^\tau = \arg \min_{\beta} \sum_{l \in \mathcal{L}, t+l < S} (\rho_\tau(f(Y_{itL}) - X_{itl}\beta)) + \lambda \|\beta\|_1$$

where $\mathcal{L}(l') = \{l : l' - c \leq l \leq l' + c\}$. The length lag window for this training is $2c + 1$. Shortening the training window is not computationally advantageous since more models need to be trained, but the projection performance will be changed.

Adjusting weights of Observations The observations are not equally useful when estimating the backfill patterns. Weights are leveraged to observations as follow:

$$\beta^\tau = \arg \min_{\beta} \sum_{l \in \mathcal{L}, t+l < S} w(\rho_\tau(f(Y_{itL}) - X_{itl}\beta)) + \lambda \|\beta\|_1$$

where $w = \exp(-\gamma \cdot D_s \cdot D_y)$. D_d and D_y are two variables that affect weighting from different aspects. For each observation y_{itl} , $D_d = S - (t + l)$ measures the *freshness*, $D_y = \tilde{y}_{i,(S),0} - \tilde{y}_{itl}$ measures the similarity to the most up-to-date report. The hyper-parameter $\gamma \geq 0$ controls the additional *focus* that we put on recent data.

In general, there are three hyper-parameters to tune in practice: 1) c for controlling the training lag window ; 2) λ for the lasso penalty; 3) γ for controlling the weights of observations. Figure X1 and X2 shows the (not that confident about how to explain this part or are the figs currently good enough?)

3.2 Incorporating Fraction Adjustment for Fraction Projection

Turning our focus now to fraction projection. Different from counts projection where zeros are simply filled with missingness when there is no update provided yet, the edge cases for fraction projection is more complicated to dealing with. We mainly suffer from two cases: 1) fraction with extremely small denominators (can even be zeros) which means the revision of numerators is faster than that of denominators. This can result in incredibly large fraction observations. Such fractions are not statistically representative and mostly exist when lag is super small; 2) fraction equals to zero. The

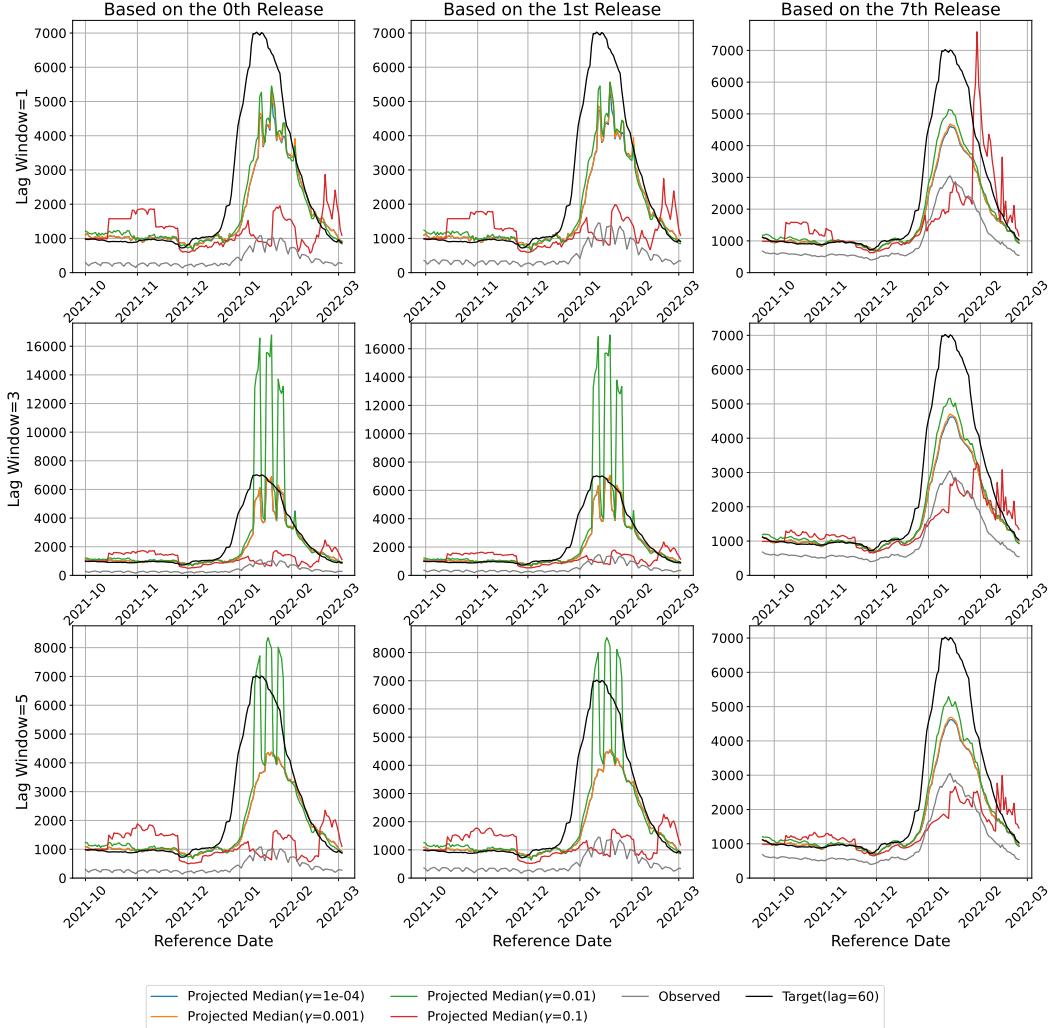


Figure 5:

log transformation $f(\cdot)$ proposed in the previous section are no longer suitable for this case since fractions are already small enough. Inappropriately large constant a can introduce noise into the data[16].

Here we propose a Beta prior approach based on a lag^{-1} model, which can also serve as a baseline model. To make all the observations valid fractions, we add 1 and 100 respectively to the numerators and denominators beforehand. For location i , at test date S , consider the model below

$$Q_{f(Y_{itL})|f(X_{itl})}^{\tau} = \beta^{\tau T} X_{itl}$$

Covariates included in X_{itl} are the following:

- $\mathbb{1}(\text{wd}(t + l) == k)$: indicator functions checking the day-of-week of the report date $t + l$, where $k \in \{0, 1, 2, \dots, 5\}$.
- $\mathbb{1}(\text{wd}(t) == k)$: indicator functions checking the day-of-week of the report date t , where $k \in \{0, 1, 2, \dots, 5\}$.

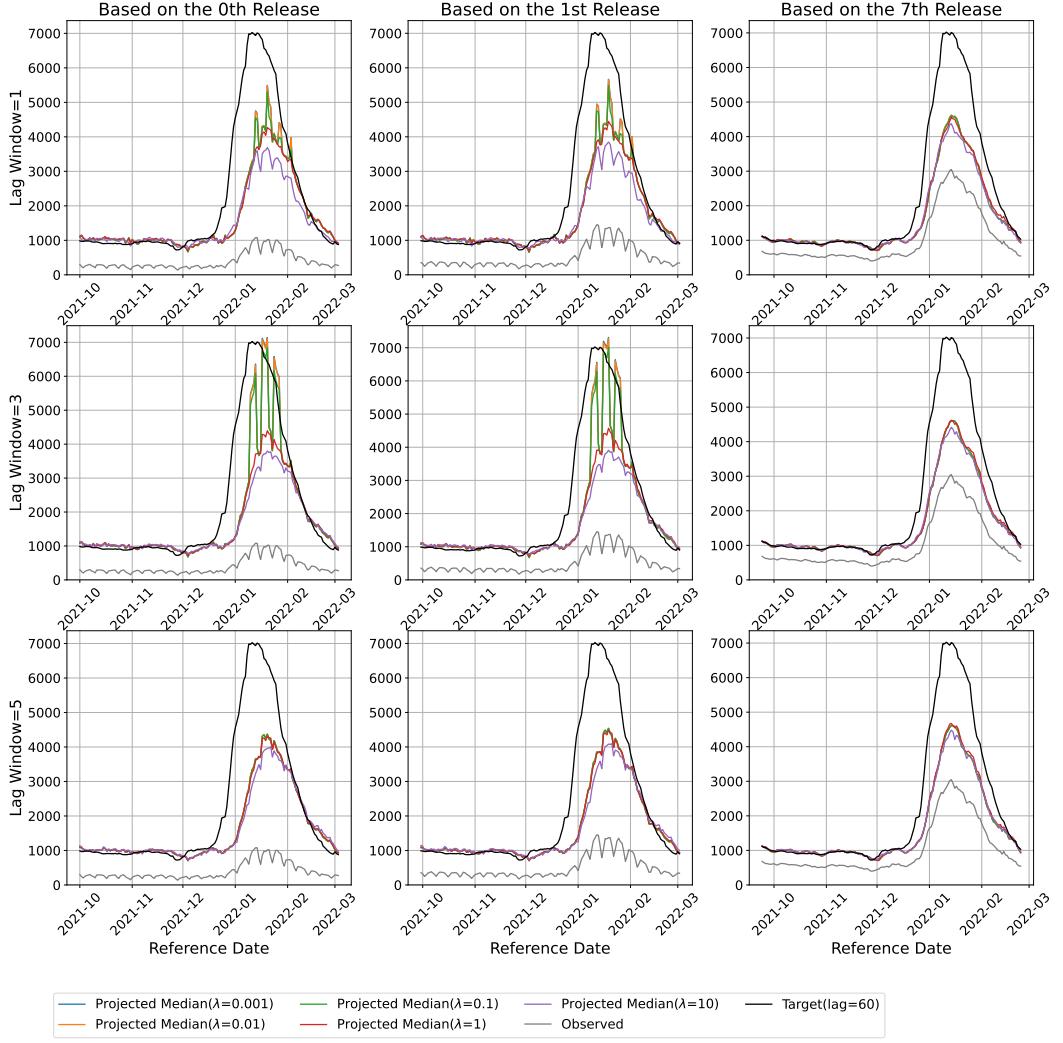


Figure 6:

- $\mathbb{1}(\text{mw}(t + l) == k)$: indicator functions checking the week-of-month of report date $t + l$, where $k \in \{0, 1, 2, \dots, 3\}$
- $f(\tilde{y}_{itl})$: average of 7-day past values on log scale.

We assume that $Y_{itl}|X_{itl}$ follows Beta distribution that are day-of-week specific. For each day of a week, the predicted quantiles of y_{itl} on log scale are exponentiated and utilized to estimate the parameters α and β of the Beta distribution using Non-Linear Minimization[17][18]. And then,

$$\text{Pseudo Numerator} = \alpha$$

$$\text{Pseudo Denominator} = \alpha + \beta$$

$$y_{itl} = \frac{\text{num}_{itl}}{\text{denom}_{itl}} \rightarrow \frac{\text{num}_{itl} + \alpha}{\text{denom}_{itl} + \alpha + \beta}$$

This adjustment is applied not only to y_{itl} but also to all corresponding covariates.

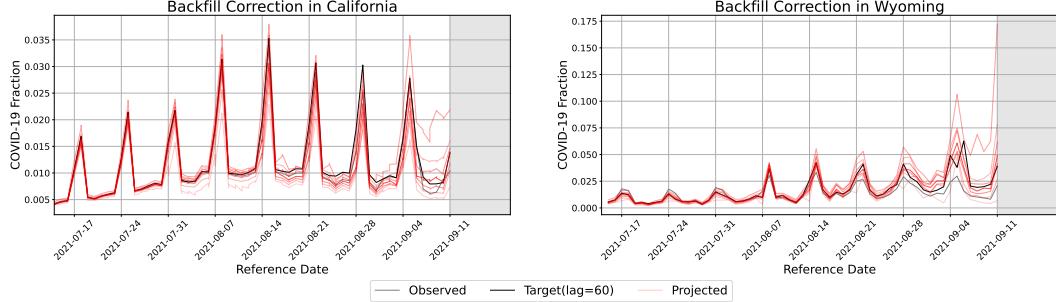


Figure 7: Illustration of estimating target of CHNG outpatient COVID-19 fractions as of 2021-09-11. The gray curve shows the most recent release of the COVID fractions based on CHNG outpatient data in California and Wyoming. The red lines show the projected quantiles of the 60th revision of the COVID-19 fractions for each reference date. The black curve shows the actual 60th revision for the values of each reference date in the future.

Figure 8:

4 Evaluation

We now evaluate projection performance for all locations of interest (50 states and top 200 populous counties) and for all test dates between 2021-08-30 and 2023-03-11 inclusive (589 days in total) against the target values as defined in section 2.1. To reiterate, in our real-time backfill correction experiment, we restrict our access to data that would have been available at each test date S . In order to save time computationally, we train the model for every 14 consecutive days but not daily.

We use the weighted interval score(WIS) to evaluate the distance between the projected distribution and the target value. Suppose our projected values are notated as $\hat{Q}_{(Y_{itl})}(\tau)$ after exponentiation, which represents the projected τ th quantile of Y_{itL} given y_{itl} and corresponding covariates. Notice that in the count projection, for each location i and reference date t pair, we have multiple data points with different lags to project the same Y_{itL} . To distinguish the projection results from these points, we will re-write them as $\hat{Y}_{itl}|X_{itl}$. Therefore, the WIS scores are calculated by comparing the distribution of \hat{Y}_{itl} on log scale with the distribution Y_{itL} on log scale no matter for counts or fractions.

Figure X shows an example of count projection based on CHNG outpatient COVID claims in California. The real-time report, which will be revised in the next two months, largely deviates from the truth temporarily. In such cases, our model can avoid misjudgment of trends due to claims counts that are not reported in time. The distribution of fractions is much different from the previous one, and the pattern is more complicated since the revisions of the fractions are no longer monotonically increasing. Figure X shows the example of fraction projection based on CHNG outpatient COVID-19 fractions in Harris County, TX. With only a limited number of revisions, the predictions provided by our model are close enough to the target values, which are available two months later after the first release.

To have a broad view of the projection performance, the projection errors are aggregated into an average on log scale and can be interpreted as the geometric average of relative error if exponentiated. The real-time observation can roughly serve as the projected median from a baseline model. For the purposes of making fair comparisons, we include the mean absolute error on log scale using the projected media

Test on other datasets: choice 1) MA DPH : well known, but not available for all dates of interest (starting from 2021, 1, 4. End of 2022, 7, 8) choice 2) Quidel Antigen Data choice 3) CHNG Inpatient Data

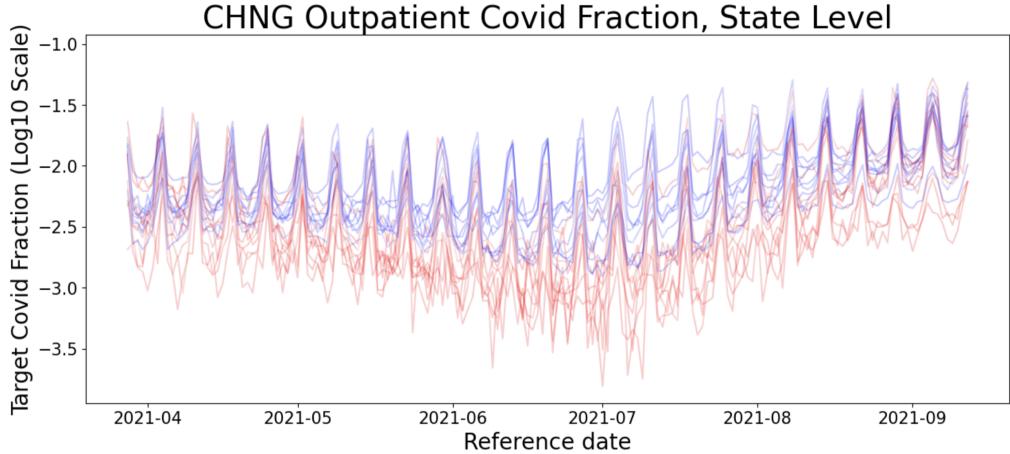


Figure 9: red lines represent the 10 states with the worst prediction performance while blue lines represent the 10 states with the best prediction performance

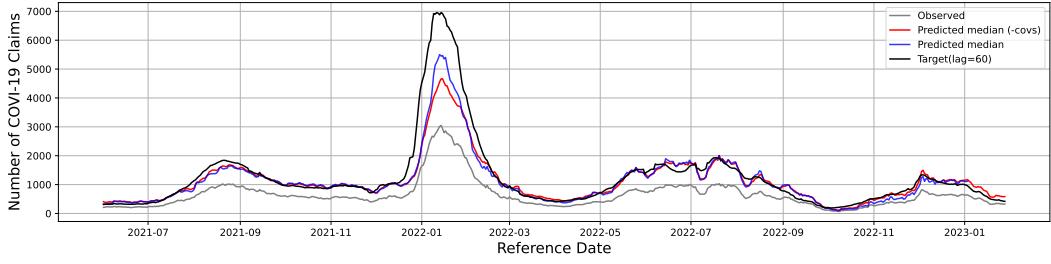


Figure 10: the backfill correction

5 Discussion

The curve of %percent of reported counts over revisions indicates that specific backfill patterns exist. Quantification of such backfill patterns and the evaluation of projection results remains a challenge. We use weighted interval scores to evaluate our log-scale projection to investigate the relative errors which can vary broadly in different regions and time periods. The current strategy is reasonable enough but not perfect. As shown in Figure X shows examples of poor projection performance for the locations with small target fractions. Minor absolute errors are magnified in the relative error evaluation system. We believe that the same relative error is less acceptable for areas with higher infection densities.

Due to the exponential nature, situations in which models missed the beginning of upswings are more strongly emphasized while failing to predict a downturn following a peak is less severely penalized.

6 Acknowledgements

The authors are grateful to Aaron Rumack, Roni Rosenfeld, Bryan Wilder, Ryan Tibshirani, Logan Brooks, Will Townes for insightful conversations. This work was supported by the fellowship from the Center for Machine Learning and Health at Carnegie Mellon.

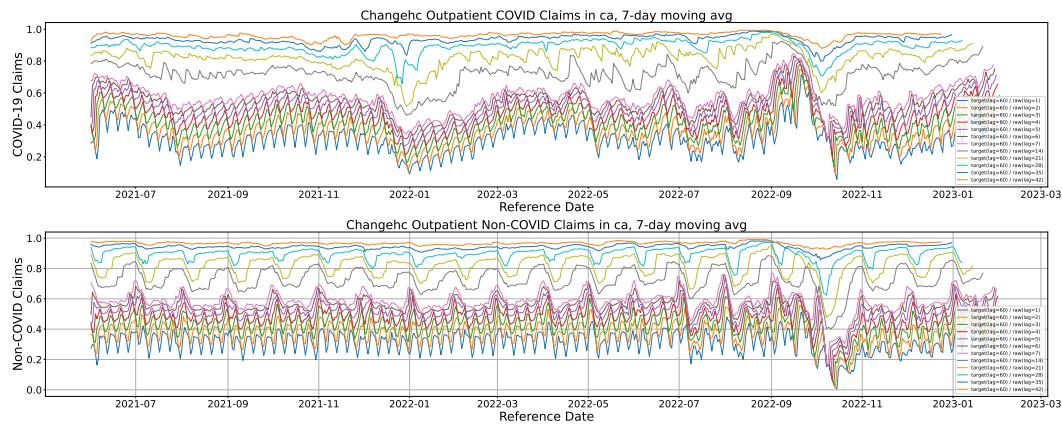


Figure 11: *the backfill correction*

References

- [1] Sarah F. McGough et al. “Nowcasting by Bayesian Smoothing: A flexible, generalizable model for real-time epidemic tracking”. In: *PLOS Computational Biology* 16.4 (Apr. 2020). Ed. by Bryan Lewis, e1007735. DOI: 10.1371/journal.pcbi.1007735. URL: <https://doi.org/10.1371/journal.pcbi.1007735>.
- [2] Roni Rosenfeld and Ryan J. Tibshirani. “Epidemic tracking and forecasting: Lessons learned from a tumultuous year”. In: *Proceedings of the National Academy of Sciences* 118.51 (Dec. 2021). DOI: 10.1073/pnas.2111456118. URL: <https://doi.org/10.1073/pnas.2111456118>.
- [3] Nicholas G Reich et al. “A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the United States”. In: *Proceedings of the National Academy of Sciences* 116.8 (2019), pp. 3146–3154.
- [4] Prithwish Chakraborty et al. “What to know before forecasting the flu”. In: *PLOS Computational Biology* 14.10 (Oct. 2018). Ed. by Marcel Salathé, e1005964. DOI: 10.1371/journal.pcbi.1005964. URL: <https://doi.org/10.1371/journal.pcbi.1005964>.
- [5] Prithwish Chakraborty et al. “What to know before forecasting the flu”. In: *PLoS computational biology* 14.10 (2018), e1005964.
- [6] Prashant Rangarajan, Sandeep K Mody, and Madhav Marathe. “Forecasting dengue and influenza incidences using a sparse representation of Google trends, electronic health records, and time series data”. In: *PLoS computational biology* 15.11 (2019), e1007518.
- [7] Alexander Rodriguez et al. “Deepcovid: An operational deep learning-driven framework for explainable real-time covid-19 forecasting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 17. 2021, pp. 15393–15400.
- [8] JF Lawless. “Adjustments for reporting delays and the prediction of occurred but not reported events”. In: *Canadian Journal of Statistics* 22.1 (1994), pp. 15–31.
- [9] Michael Höhle and Matthias an der Heiden. “Bayesian nowcasting during the STEC O104:H4 outbreak in Germany, 2011”. In: *Biometrics* 70.4 (2014), pp. 993–1002.
- [10] Logan C Brooks et al. “Nonmechanistic forecasts of seasonal influenza with iterative one-week-ahead distributions”. In: *PLoS computational biology* 14.6 (2018), e1006134.
- [11] Tenglong Li and Laura F White. “Bayesian back-calculation and nowcasting for line list data during the COVID-19 pandemic”. In: *PLoS computational biology* 17.7 (2021), e1009210.
- [12] Maria Jahja, Andrew Chin, and Ryan J. Tibshirani. *Real-Time Estimation of COVID-19 Infections: Deconvolution and Sensor Fusion*. 2021. DOI: 10.48550/ARXIV.2112.06697. URL: <https://arxiv.org/abs/2112.06697>.
- [13] Harshavardhan Kamarthi, Alexander Rodriguez, and B Aditya Prakash. “Back2Future: Leveraging backfill dynamics for improving real-time predictions in future”. In: *arXiv preprint arXiv:2106.04420* (2021).
- [14] Alex Reinhart et al. “An open repository of real-time COVID-19 indicators”. In: *Proceedings of the National Academy of Sciences* 118.51 (2021), e2111452118.
- [15] Roger Koenker and Gilbert Bassett. “Regression Quantiles”. In: *Econometrica* 46.1 (Jan. 1978), p. 33. DOI: 10.2307/1913643. URL: <https://doi.org/10.2307/1913643>.
- [16] Nikos I. Bosse et al. “Scoring epidemiological forecasts on transformed scales”. In: (Jan. 2023). DOI: 10.1101/2023.01.23.23284722. URL: <https://doi.org/10.1101/2023.01.23.23284722>.
- [17] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. 1st ed. Englewood Cliffs, New Jersey 07632: Prentice-Hall, 1983.
- [18] Robert B. Schnabel, John E. Koonatz, and Barry E. Weiss. “A modular system of algorithms for unconstrained minimization”. In: *ACM Transactions on Mathematical Software* 11.4 (Dec. 1985), pp. 419–440. DOI: 10.1145/6187.6192. URL: <https://doi.org/10.1145/6187.6192>.