Modeling Epidemiological Time Series

Aaron Rumack

August 2023
CMU-ML-23-105

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee

Roni Rosenfeld, chair
Ryan Tibshirani
F. William Townes
Marc Lipsitch (Harvard)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by: Centers for Disease Control and Prevention award numbers U01IP001121, U01IP001141 and 75D30123C15907; Defense Threat Reduction Agency award number HDTRA118C0008; National Institutes of Health award number U54GM088491; McCune Foundation award number 2020049601; and a graduate fellowship from the University of Pittsburgh Medical Center Enterprises.

Keywords: epidemiological forecasting, epidemiological modeling, probabilistic forecasting, calibration, influenza, COVID-19, insurance claims, signal heterogeneity

Abstract

Epidemiological data present naturally in a time series format. In many different contexts, these time series have structured biases due to the nature of the data generating processes. Identifying and correcting for these biases is crucial for accurate epidemiological modeling and forecasting. However, bias correction is a difficult task because biases vary by data source, there is limited access to historical data, and ground truth labels are almost always unavailable. In this thesis, we look at two classes of epidemiological time series: forecasts and real-time "indicators" of disease activity. For both classes, we describe the process of identifying biases and present different algorithms to correct them, depending on the context. We provide applications in modeling and forecasting influenza and COVID-19.

Acknowledgements

First, I would like to thank Roni, my advisor. He truly has been an advisor to me throughout my years in the program, not only in research and statistics, but also in social responsibility, professional relationships, mentoring, and life in general. He patiently guided me through many projects, and taught me how to approach problems and see the bigger picture - both conceptually and in relation to the real world. I have been very fortunate to work with him.

I am also grateful to my committee for their guidance and feedback, which has greatly improved the quality of the results presented in this thesis. Ryan has been an inspiration to me in sheer intellectual force and work ethic, delivered with true kindness. Will greatly expanded my statistical toolbox and gave me a new perspective on problem solving. Marc connected the data and results to real-world phenomena and served as an important reminder of the applications of my research.

I am indebted to the Delphi group as a whole, which grew exponentially during my time at CMU, to the extent that I cannot thank everyone individually. I want to specifically thank Logan Brooks, Maria Jahja, Katie Mazaitis, Jingjing Tang, Ananya Joshi, and Xueda Shen who collaborated with me over the years. I also want to thank Chirag Gupta and Youngseog Chung for giving me another opportunity to work on forecast calibration. All of them have made my time at Carnegie Mellon more enjoyable and fruitful.

I am grateful to the various agencies and foundations that have supported this research financially: the Centers for Disease Control and Prevention (CDC), Defense Threat Reduction Agency (DTRA), National Institutes of Health (NIH), McCune Foundation, and University of Pittsburgh Medical Center Enterprises. The CDC supported this research in a practical way by initiating the FluSight Challenge and motivating a collaborative environment of influenza forecasting. Optum and Change Healthcare shared large datasets with the Delphi group, which allowed for detailed analysis of influenza and COVID-19 trends.

None of this would have been possible without the support, encouragement, and guidance of friends and family. My father was my first and most formative teacher in math, science, and life, and it is because of him that I was able to reach this point. My mother has always provided me unconditional love, support, and advice, which has been crucial for me in everything that I do. My wife, Annette, has stood by my side unwaveringly throughout my PhD, sharing in the good and bad moments, and I cannot imagine these years without her. My brother, Samuel, has always told me the truth I needed to hear, which encouraged me to continue working until I reached me goals. I also want to thank the Squirrel Hill community as a whole, as well as individuals far too numerous to name individually, for ensuring that my years in Pittsburgh were ones of personal as well as academic growth.

Lastly, I thank Hashem, the One who spoke and created the world, a world full of information, patterns, and structure. I hope that this research constitutes a small part of my duty to understand these patterns and contribute positively to the best of my ability.

Contents

Chapter 1

Introduction

1.1 Motivation

Infectious diseases are a severe burden both in medical and economic terms. In 2019, lower respiratory infectious were the fourth leading cause of death globally and three of the top ten leading causes of death were communicable diseases [38]. Advances in public health and medicine have ameliorated the effects of infectious diseases, but they still remain quite deadly.

Modeling infectious disease epidemics, such as influenza and COVID-19, can allow public health officials to better understand the current situation and implement more effective measures against the spread of the disease. Accurate forecasts can further aid the public health response. Modeling and forecasting, if accurate and used properly, have the potential to reduce the number infections during an epidemic, mitigate the average case severity, and reduce the overall burden on the healthcare system.

Machine learning and statistical models perform best when trained on copious and reliable data. Even in fields where data is relatively plentiful, modelers face challenges with biases in real-world data. In epidemiology, where data is relatively sparse, these challenges can hinder the accuracy of models and forecasts. The U.S. Centers for Disease Control and Prevention (CDC) has initiated several surveillance programs for selected infectious diseases, including ILINet for influenza. The FluSight Challenge solicits forecasts of influenza-like illness (ILI) and evaluates them against observations from ILINet. Volunteer health providers submit the total number of patients seen during a given week, as well as the number of patients for ILI during that week. These reports are aggregated at the state level and released to the public.

Health insurance claims provide an auxiliary data source for disease surveillance. Health insurance providers have access to enormous quantities of data related to interactions with the healthcare system. Insurance claims data can provide information similar to ILINet in motivation, but at a larger magnitude and finer resolution. Despite these advantages, it is also subject to several biases. The raw signal of fraction of influenza (or another specified condition) visits is a function of not only influenza activity, but also day-of-week effects, holiday effects, and within-month effects. In order to make the signal most useful for modeling and forecasting, we must correct for these effects.

Other auxiliary data sources can be used as indicators of disease activity and incorporated into a real-time system to understand the current disease burden. These data sources

could include insurance claims but also internet search behavior, cell phone mobility, and surveys [44]. These auxiliary data sources have several benefits relative to traditional public health surveillance data. They are generally available with larger sample sizes, less latency, and at different severity levels. These signals can be useful for forecasting because they often lead targets of interest such as outpatient or inpatient visits.

Each of these various data sources have different strengths and weaknesses. In general, the utility of a data source is measured by geographic granularity, temporal granularity, coverage, and latency. Granularity is defined by the size of the locations and time periods at which the data is aggregated or averaged. We prefer datasets with fine granularity, as they contain more information than those with coarse granularity. Coverage refers to the proportion of the total population represented in the data, with the dataset's utility increasing with increasing coverage. Finally, latency refers to the delay at which the data is available. In many applications, we wish to model and forecast in real-time, so a dataset is more useful if it is available and accurate with a lower latency.

Traditional surveillance systems such as ILINet usually fare poorly by these metrics. ILINet is aggregated at a state and week granularity, and coverage is low as the network consists solely of volunteer providers. Health insurance claims can be aggregated at a much finer spatial and temporal scale, potentially at the 5-digit ZIP Code and daily level. Depending on the provider, coverage can be much higher as well. However, both of these sources suffer from significant latency. Digital surveillance sources are often available at fine granularity, high coverage, and low latency, making them highly desirable for integration into real-time forecasting and modeling systems. The downside to digital surveillance sources is that they are often less specific to a given disease or condition. For example, it is often difficult to distinguish between different respiratory infections (or detect whether an individual is infected at all) based on internet search queries.

Whether disease indicators are derived from traditional surveillance systems or auxiliary data sources, they measure a target that is merely related to the actual disease activity. Therefore, there will almost always be biases in the relationship between the indicator and target. These biases vary between locations and across time, and will obviously limit the benefits of using these signals. If we display a map of an indicator that suffers from spatial heterogeneity, the map may show one location with much higher indicator values than another location, when in reality, the other location has a higher disease activity. An indicator that suffers from temporal heterogeneity will similarly be misleading and lead to inaccuracies in forecasts. Real-time indicators can be useful for forecasting and public health officials, but will be most useful when they more directly relate to the target of interest.

Biased or insufficient data can lead to inaccurate forecasts. Forecasts can also be inaccurate due to erroneous models. The FluSight Challenge forecasts can be mostly divided into two groups: mechanistic models and statistical models. Mechanistic models generally use domain knowledge to model disease transmission and simulate the future trajectory of an epidemic. One approach to mechanistic modeling is the use of compartmental models, where the entire population is broken down into compartments based on their disease state. A common example of a compartment model is the SIR model [30], which divides the population into susceptible, infectious, and recovered compartments and uses differential equations to describe the movement of individuals between the compartments over time. Agent-based modeling is another approach, which often uses simulations to model individual behavior and disease transmission. Statistical models generally avoid relying on domain knowledge and treat the forecasting task as a time-series prediction task. Mechanistic models can fail when the underlying model diverges too strongly from reality, and statistical models can

fail when unique events occur and the time-series model cannot generalize to a new testing regime. Each individual model can fail in its own way, but often these failures are systematic and can be considered as biases. Identifying and correcting for these biases improves the accuracy and utility of the forecasts.

In the FluSight Challenge, the CDC solicits distributional forecasts, where the forecaster must produce a probabilistic distribution of the target variable, rather than merely outputting the forecaster's predicted mean or median of the target. It is more difficult to produce accurate distributional forecasts than point forecasts. First, a point forecast is usually a function of a distributional forecast, meaning that a distributional forecast contains strictly more information. Additionally, models are often less accurate in modeling the tails of a distribution due to the scarcity of observations. Point forecasts are also more studied and have a longer history of development. Calibration is a measure of reliability and an important quality of distributional forecasts. If forecasts are not calibrated, then those who use those forecasts will make decisions based on inaccurate information.

Biases in forecasts, disease surveillance data, and indicators all prevent epidemiological modeling from being applied optimally to reduce the disease burden. By identifying and correcting these biases, we can use our data more effectively to better understand and respond to epidemics.

1.2 Overview

This thesis describes three examples in which we corrected biases in infectious disease data sources and models.

In Chapter 2, we describe calibration as applied to continuous forecasts and demonstrate that many models submitted to the FluSight Challenge are significantly miscalibrated. We present a recalibration method that works on any black-box forecaster by estimating the distribution of probability integral transform (PIT) values. We improve the method by accounting for special challenges in the domain of epidemic forecasting, such as seasonality. We also discuss theoretical links between calibration and proper scoring rules.

In Chapter 3, we describe a large health insurance claims dataset and the biases present in the raw signal. These biases are the result in temporal changes to healthcare-seeking behavior and in healthcare availability. If these biases are left uncorrected, the resulting naive signal leads to false conclusions, especially regarding the timing of an influenza season's peak. We present a method to correct for these biases and produce a smooth signal of underlying disease activity that can be used to model the spatiotemporal dynamics of the disease. We also present a validation method to demonstrate that the bias correction is indeed effective.

In Chapter 4, we describe the problem of spatial and temporal heterogeneity between two signals, one of which has systematic biases. We present a method to use a "guiding" signal to correct for these biases and produce a more reliable signal that can be used for modeling and forecasting. The method assumes that the heterogeneity can be approximated by a low-rank matrix and that the temporal heterogeneity is smooth over time. We also present a hyperparameter selection algorithm to choose the parameters representing the matrix rank and degree of temporal smoothness.

We conclude with Chapter 5, which summarizes the main results of the thesis and suggests directions for future work.

Chapter 2

Recalibrating Probabilistic Forecasts

Much of this chapter is based on [46].

2.1 Introduction

Epidemic forecasting is an important tool to inform the public health response to outbreaks of infectious diseases. Often, decision makers can take more effective action with an estimate of the uncertainty in a forecasted target. For this reason, distributional forecasts are more desirable than point forecasts. A distributional forecast is a probability distribution over the target variable and measures the uncertainty in the prediction, as opposed to a point forecast, which is just a scalar value for each target and has no measure of uncertainty. A desired property of distributional forecasts is *calibration*, or reliability between forecasts and the true distribution of the variable forecasted (a mathematical definition is given in Section 2.2). Along with uncertainty and resolution, calibration is one of three components of a forecaster's accuracy as measured by any proper score [4], with better calibration resulting in a better score. It is therefore important for a forecaster to produce calibrated forecasts.

Previous work has described general forecasting theory and calibration and evaluated the calibration of certain forecasts [12, 18, 19, 28]. Later work has gone beyond just describing calibration, presenting post-processing algorithms to recalibrate forecasts that were previously miscalibrated. Nonparametric techniques for recalibration of ensemble forecasts include rank histogram correction [24], Bayesian model averaging [41], linear pooling [20], and probability anomaly correction [51]. Brocklehurst et al. [5] provide a nonparametric approach using the empirical CDF, which can recalibrate any forecast of a scalar target. Parametric approaches include logistic regression [26], extended linear regression [21] and beta-transform linear pooling [20]. Wilks and Hamill [54] compare the performance of different recalibration techniques for different meteorological targets with different amounts of training data.

Much of the work in recalibration has been applied to weather forecasting, and thus many of the techniques are not applicable in other forecasting domains. The most popular weather forecasting models create a distribution from a series of point predictions, with

each point being the result of a simulation under varying initial conditions. Many of the existing recalibration methods are defined only for this type of ensemble forecaster. For example, Bayesian model averaging assumes that an ensemble forecast is comprised of the same N forecasts in each observation. This method cannot be extended trivially to a domain where the forecaster itself outputs a distribution. Additionally, weather forecasts usually have a plethora of training data on which to train recalibration methods. For example, recalibration has been applied to a set of weather forecasts generated daily from 1979 to at least 2006, almost 10,000 days [25]. In settings like these, techniques need not be robust to small amounts of recalibration training data.

To be clear on nomenclature, throughout this chapter, we use the term *forecast* to refer to the predicted probability distribution of a variable and the term *forecaster* to refer to an algorithm that produces a forecast for a variable given a context. Common examples of forecasters are an algorithm that forecasts the amount of precipitation two days in advance given current meteorological information, one that forecasts the price of a certain stock given the stock's historical trend, or one that forecasts the statewide influenza incidence given historical incidence data. We also distinguish between *calibration* and *recalibration*; calibration refers to the property of a forecaster, and recalibration refers to a method whose goal is to make a forecaster more calibrated. Specifically, recalibration takes as input a set of a forecaster's forecasts and corresponding observations ("training data"), and outputs a forecaster which should be more calibrated on a different set of forecasts and observations ("test data").

In what follows, we present a generalized approach to forecast recalibration and show its performance when applied to forecasters in the FluSight Network. We demonstrate that across the diverse set of FluSight forecasters, recalibration consistently improves not just calibration but accuracy as well.

2.2 Methods

Consider the following setup. At each $i = 1, 2, 3 \ldots$, a forecaster M outputs a density forecast f_i given features x_i for a continuously distributed scalar random variable y_i whose true distribution is h_i. As a regularity condition, we assume that the corresponding cumulative distribution functions (CDFs) F_i and H_i are continuous and strictly increasing. The forecaster M is evaluated according to a proper scoring rule, such as the quadratic score [17] or the logarithmic score [22].

The goal of a forecaster is to produce ideal forecasts, i.e., to forecast $f_i = h_i$, the true distribution of y_i, for each i, though this is usually unattainable. We can inspect how close a forecaster is to being ideal with the distribution of the probability integral transform (PIT) values [13]. For each forecast f_i and observed value y_i, the PIT is defined as

$$\mathrm{PIT}(f_i, y_i) = F_i(y_i),$$

where F_i is the CDF of f_i. A necessary (but not sufficient) condition for a forecaster to be ideal is *probabilistic calibration* [18]:

$$\frac{1}{N} \sum_{i=1}^{N} H_i \circ F_i^{-1}(p) \to p \text{ as } N \to \infty, \text{ for all } p \in (0, 1).$$

(Here and throughout we interpret convergence in the almost sure sense.) An example of a probabilistically calibrated forecaster that is not ideal is the so-called climatological

forecaster, which for each i outputs the marginal distribution of y_i over $i = 1, 2, 3, \ldots$. To make this concrete, suppose each y_i is distributed as $\mathcal{N}(\mu_i, 1)$, a normal distribution with mean μ_i and variance 1, and each μ_i itself follows $\mathcal{N}(0, 1)$, then the climatological forecaster simply outputs $\mathcal{N}(0, 2)$ for each i.

Note that the PIT distribution of a probabilistically calibrated forecaster is close to uniform in large samples. The expected CDF of the PIT distribution is

$$G(p) = \mathbb{E}[\mathbb{P}[F_i(y_i) \leq p]] = \mathbb{E}[\mathbb{P}[y_i \leq F_i^{-1}(p)]] = \mathbb{E}[H_i \circ F_i^{-1}(p)],$$

where here $\mathbb{E}$ denotes the sample average operator over $i = 1, \ldots, N$. This expression converges to p as $N \to \infty$ when the forecaster is probabilistically calibrated. Thus an examination of the distribution of PIT values—looking for potential deviations from uniformity—serves as a good diagnostic tool to assess probabilistic calibration. Many use a PIT histogram to examine the PIT distribution because it is easy to read and understand [18]. For example, if the PIT distribution is bell-shaped, then the forecaster does not put enough weight in the middle of its distribution and is underconfident. In general, we can compare the PIT density to the horizontal line at 1, which corresponds to the uniform density. The greater the deviation from this line (which can be quantified via Kullback-Leibler divergence from the uniform distribution to the PIT distribution, or equivalently, negative entropy of the PIT distribution), the greater the miscalibration; see Fig 2.1 for examples.

Our recalibration method uses G as a CDF-CDF transform. The recalibrated forecaster, denoted M^*, is defined by a recalibrated forecast CDF of $F_i^*(y) = G(F_i(y))$, for each i. By the chain rule, the recalibrated forecast density is $f_i^*(y) = g(F_i(y)) \cdot f_i(y)$, for each i. Thus the recalibrated forecast f_i^* is the original forecast f_i weighted by the PIT density g. An illustration of this method is provided in Fig 2.2. In practice, of course, we do not have access to the true distributions H_i, so we need to estimate G from PIT values. A key assumption is that the PIT distribution of the training forecasts is the same as that of the test forecasts. Otherwise, applying G as a CDF-CDF transform will not produce probabilistically calibrated forecasts. The ultimate estimate of G that we propose in this paper will be an ensemble (weighted linear combination) of three estimates: a nonparametric method, a parametric method, and a null method. First, we will motivate calibration as a tool to increase forecast accuracy, and then, we explain the individual estimation methods.

2.2.1 Calibration and log score

In order to quantify how well a forecaster is calibrated, we calculate the entropy of the distribution of PIT values. As above, G is the CDF of the PIT distribution of M. The entropy of the PIT density g is defined as

$$H(g) = -\int_{p=0}^{1} g(p) \log g(p)\, dp.$$

If M is probabilistically calibrated, then (asymptotically, as $N \to \infty$) the PIT values are uniform and the entropy is zero because $g(p)$ is 1 everywhere. When the PIT values are not uniform, the entropy is negative.

Entropy is also useful because it provides an understanding of how miscalibration penalizes the expected log score, as shown below. First observe that

$$g(p) = \frac{d}{dp} G(p) = \frac{d}{dp} \mathbb{E}[H_i \circ F_i^{-1}(p)] = \mathbb{E}\left[\frac{h_i(F_i^{-1}(p))}{f_i(F_i^{-1}(p))}\right],$$

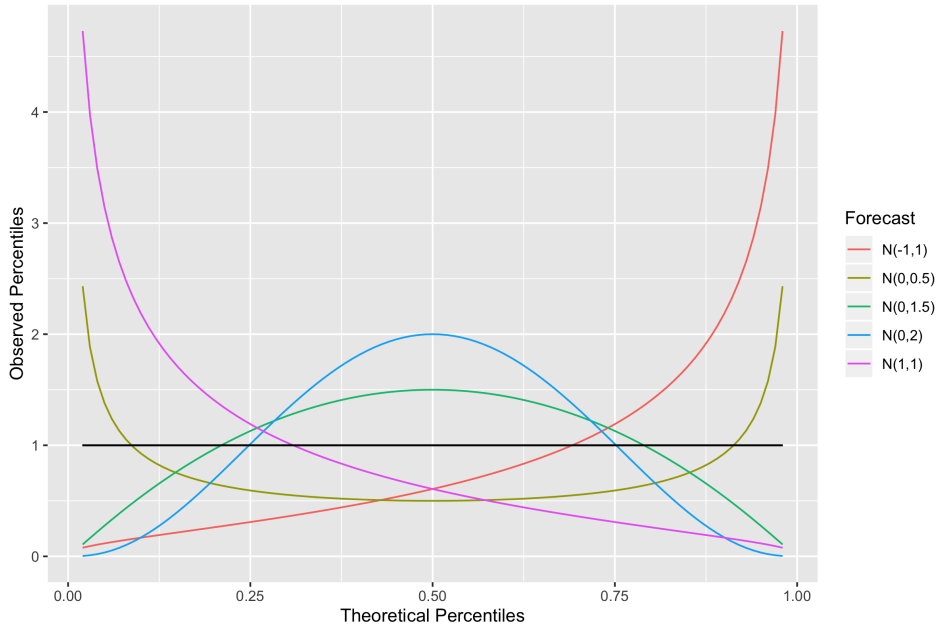

Figure 2.1. Densities of PIT distributions for five sample forecasters, when the true distribution is a standard normal.

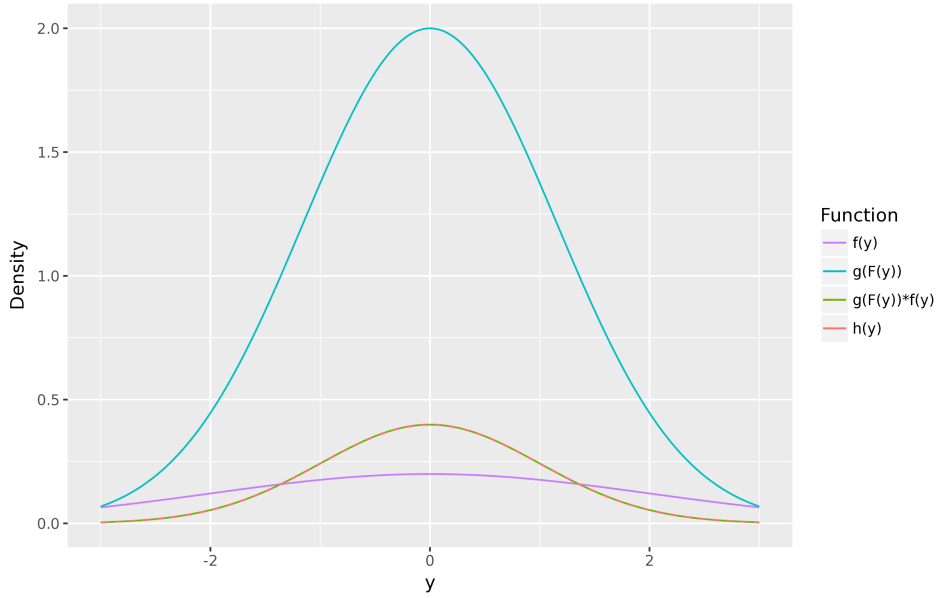

Figure 2.2. An illustration of recalibration. The original, underconfident forecast density is $f(y) = \mathcal{N}(0, 2)$ while the true density is $h(y) = \mathcal{N}(0, 1)$. By calculating the PIT density g and producing a recalibrated forecast as the product $g(F(y)) \cdot f(y)$, we recover the true $h(y)$.

where the last step assumes the smoothness and integrability conditions on h_i, f_i needed to exchange expectation and differentiation (the Leibniz rule). Next observe that

$$\begin{aligned}
\mathbb{E}[\log f_i^*(y_i)] - \mathbb{E}[\log f_i(y_i)] &= \mathbb{E}[\log g(F_i(y_i))] \\
&= \mathbb{E}\left[\int_{-\infty}^{\infty} \log g(F_i(y_i)) h_i(y_i)\, dy_i\right] \\
&= \mathbb{E}\left[\int_0^1 \log g(F_i(F_i^{-1}(p))) \frac{h_i(F_i^{-1}(p))}{f_i(F_i^{-1}(p))}\, dp\right] \\
&= \int_0^1 \mathbb{E}\left[\log g(p) \frac{h_i(F_i^{-1}(p))}{f_i(F_i^{-1}(p))}\right] dp \\
&= \int_{p=0}^1 g(p) \log g(p) = -H(g), \qquad (2.1)
\end{aligned}$$

where the third line is obtained by a variable substitution, and fourth by applying the Leibniz rule again assuming the needed regularity conditions.

For any forecaster, if the PIT distribution is the same for the training data and the test data, then the improvement of the recalibrated forecast's log score can be estimated by estimating the negative entropy of g (note that the entropy of any distribution on $[0,1]$ is nonpositive). We can explain this intuitively as well: the more negative $H(g)$ is, the more it indicates that there is information lying in the structure of g that can be extracted to improve forecasts.

2.2.2 Nonparametric correction

Given an observed training set of PIT values for a forecaster, $F_i(y_i)$, $i = 1, \ldots, N$, the empirical PIT CDF is

$$\hat{G}(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[F_i(y_i) \leq x].$$

As $\hat{G}$ is discrete, it does not admit a well-defined density, and hence to use this for recalibration we can first smooth $\hat{G}$ using a monotone cubic spline interpolant, and then it will have a bonafide density $\hat{g}$, which is itself smooth (twice continuously differentiable, to be precise). Using this for recalibration produces $f_i^*(y) = \hat{g}_i(F_i(y)) \cdot f_i(y)$.

In practice, with a large amount of training data, recalibration using the empirical CDF as described above can be effective. However, with little training data, or a lot of diversity within the training data among the distributions of y_i, it can be ineffective for assuring calibration on the test set. This is in line with the practical difficulties of using nonparametric, distribution-free methods in general.

2.2.3 Parametric correction

Gneiting and Ranjan [20] present a recalibration method originally motivated by redistributing weights on the components of an ensemble forecast, but their method can applied generally to recalibrate any black box forecaster. Given an observed training set of PIT values, $F_i(y_i)$, $i = 1, \ldots, N$, we fit a beta density $\hat{g}$ via maximum likelihood estimation. This

in fact corresponds to the beta transform that maximizes the log score of the recalibrated forecaster on the training data [20].

This parametric model is more resilient to minimal training data, and a beta distribution is usually an effective estimate of the PIT distribution: because a beta density can be either convex or concave, it is flexible enough to fit the PIT distribution of overconfident and underconfident forecasters; and because the mean can be in the interval $(0, 1)$, it can fit biased forecasters as well. However, problematic behaviors arise at the tails. Except in exceptional cases (one or both of its two shape parameters is exactly 1), the beta density is 0 or ∞ at the endpoints of its support, which can cause problems for recalibration (there can be a big gap between the true PIT density and $\hat{g}$ in the tails).

2.2.4 Null correction

The final component of the recalibration ensemble is a null correction, in which there is no recalibration at all, i.e., we simply set $f_i^*(y) = f_i(y)$. This prevents overfitting and decreases variance of the overall ensemble correction, to be described next.

2.2.5 Recalibration ensemble

The final recalibration system uses the three components described previously and weights them in an ensemble. The ensemble weights are calculated to maximize the overall log score. Letting f_{ij}^* denote the forecast density for sample i and component j, the weights ensemble w are defined by solving the optimization problem:

$$\underset{w}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{j=1}^{p} w_j f_{ij}^*(y_i)\right) \quad \text{subject to} \quad w \geq 0,\ \sum_{j=1}^{p} w_j = 1, \tag{2.2}$$

where p is the number of ensemble components (for us, $p = 3$) and the constraint $w \geq 0$ is to be interpreted componentwise.

A component's weight in the ensemble is not necessarily proportional to that component's performance. For example, if the two best components are very similar to each other, one may have a very small weight because that component's information is effectively represented by the other component.

2.2.6 Recalibration under seasonality

Epidemic forecasting presents a new challenge for recalibration. The methodology discussed above assumes that the previous behavior of a forecaster is indicative of future behavior, or more concretely, that the PIT distribution on the training set will be similar to that on the test set. However, this is not necessarily the case in epidemic forecasting, due to the fact that a forecaster's behavior generally changes across the different phases of an epidemic. For example, some forecasters do not predict enough of a change in disease incidence from one week to the next. For such a forecaster, the PIT values are usually too high between a season's onset and peak, because incidence increases more quickly than forecasted. Conversely, after the season peaks, the PIT values are too low, because incidence decreases more quickly than forecasted.

In order to account for such nonstationarity in the PIT distribution, we would like to form and use a special training set based on forecasts made at similar points in the epidemic

curve in different seasons. This is not a straightforward task to do in real-time, since one cannot always be sure whether the peak has passed yet or not. However, for seasonal epidemics, we can take advantage of seasonality and build this training set based on the calendar weeks in which the forecasts were made. For example, a forecast made in week 6 can be recalibrated based on forecasts in other seasons made in weeks in between 3 and 9. This is what we do in our experiments in this paper, with more details given in the next section.

2.3 Results

We apply this ensemble recalibration method to data from influenza forecasting in the US. In an effort to better prepare for seasonal influenza, the US CDC has organized a seasonal influenza forecasting challenge every year since 2013, called the FluSight Challenge [7]. In 2017, a group of forecasters formed the FluSight Network [43] and began submitting an ensemble forecast of 27 component forecasters. As part of this collaboration, each of these forecasters produced and stored retrospective forecasts spanning 9 seasons, from 2010-11 to 2018-19. The retrospective forecasts were produced at the same time, with each forecaster using the same method for all seasons. Had the forecaster modified its algorithm from season to season, the previous forecast performance would not be predictive of future forecast performance, violating the assumptions behind this recalibration method. These forecasters include mechanistic and non-mechanistic forecasters, as well as baseline forecasters. They are diverse in behavior, accuracy, and calibration, and therefore provide an interesting challenge for our recalibration method, which treats the forecaster as a black box.

First, we summarize the retrospective forecasts in the FluSight data set. Each week, a forecast is produced for seven forecasting targets, all of which are based on weighted ILI (wILI), a population-weighted average of the percentage of outpatient visits with influenza-like illness derived from reports to the CDC from a network of healthcare providers called ILINet [8]. The forecasting targets are:

- season onset (the first week where wILI is above a predefined baseline for three consecutive weeks);
- season peak week (week of maximum wILI);
- season peak percentage (maximum wILI value);
- the wILI value at 1, 2, 3, and 4 weeks ahead of the current week.

The first three targets are referred to as seasonal targets and the last four targets are referred to as short-term targets. Each forecast is discretized over predetermined bins, forming a histogram distribution. For the season onset and season peak week targets, the width of each bin is one week, and for the other targets, the width of each bin is 0.1% wILI. Forecasts are produced for each of the 10 HHS Regions as well as the US as a whole, for a total of 9 seasons, from 2010-11 to 2018-19. Thus to be clear, the forecasts in this FluSight data set are indexed by forecaster, target, season, forecast week, and location.

Next, we describe the training setup we use for recalibrating the forecasts in this data set, which is a kind of nested leave-one-season-out cross-validation. This is laid out in the steps below for a given forecaster and forecasting target, and a particular season s.

1. Create recalibrated forecasts for all seasons $r \neq s$, using each of the three methods: nonparametric, parametric, and null. For a forecast in season r at week i and at location ℓ, we build a training set using PIT values from all seasons other than r and s, all available forecast weeks in $[i-3, i+3]$ (within three weeks of i), and all locations. These recalibrated forecasts are only used for training the ensemble weights in the following step.

2. Optimize the ensemble weights w by solving (2.2) using the recalibrated forecasts from Step 1.

3. Create recalibrated forecasts for season s, again using each of the three methods: nonparametric, parametric, and null. This is just as in Step 1, except we use one more season in the training set. Explicitly, for a forecast in season s at week i and at location ℓ, we build a training set using PIT values from all seasons other than s, all forecast weeks in $[i-3, i+3]$ (within three weeks of i), and all locations.

4. Create ensemble recalibrated forecasts from season i, using the recalibration components from Step 3 and the weights from Step 2.

In what follows, we present and discuss the results. The code and data used to produce all of these results is publicly available online [47].

2.3.1 Effect of varying window size

The training procedure just presented assumes a window of $k = 3$ weeks on either side of a given week i in order to build the set of PIT values used for recalibration (using forecast data from other seasons). However, we could consider varying k, which would navigate something like a bias-variance tradeoff. We would expect the optimal window k to be larger for the nonparametric recalibration method versus the parametric one. It turns out that $k = 3$ is typically a reasonable choice for both, as displayed in Fig 2.3.

2.3.2 Forecast accuracy and calibration

For the short-term targets, the ensemble recalibration method improves the mean log score for almost all forecasters. Both the nonparametric and parametric recalibration methods significantly improve the mean log score, and the ensemble improves it even further. For the seasonal targets, some component recalibration methods do not improve accuracy, although the ensemble method does improve accuracy, averaged over all forecasters. However, the ensemble improves accuracy for seasonal targets in only about three-quarters of forecasters. See Fig 2.4 and Fig 2.5.

Fig 2.6 gives a more direct comparison of improvements in accuracy versus calibration, i.e., in mean log score versus entropy, for the short-term forecasts. (Note that we estimate the entropy of the distribution of PIT values using a simple histogram estimator with 100 equal bins along the interval $[0, 1]$.) We see a clear linear trend, with slope approximately 1, confirming our expectations from (2.1).

Finally, in Fig 2.7, we show that our ensemble recalibration method increases the entropy of the PIT distribution to nearly zero for nearly every forecaster. The two exceptions, the line segments towards the bottom of Fig 2.7, correspond to particularly poor forecasters (so poor that are outperformed by a baseline forecaster that outputs a uniform distribution).

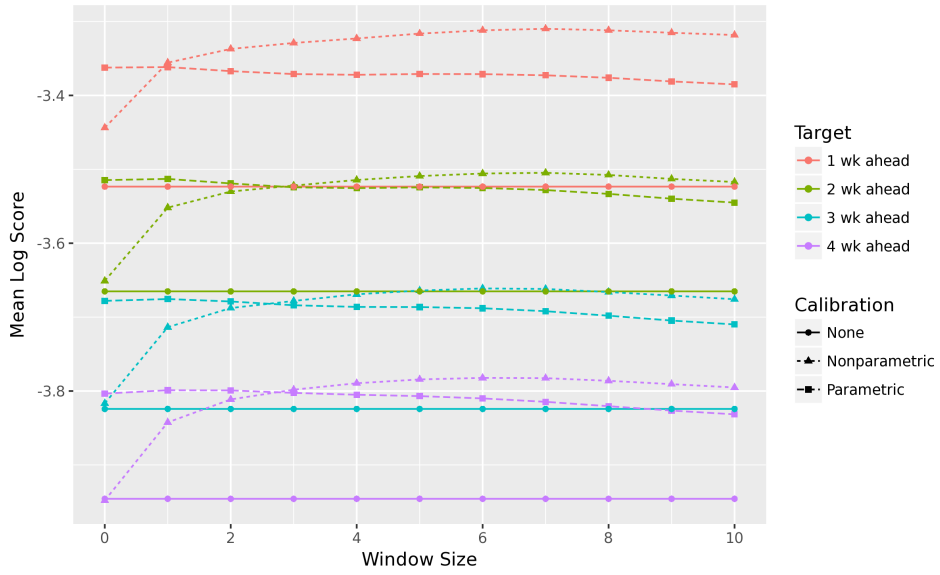

Figure 2.3. Mean log score, averaged over all forecasters, for the different recalibration methods. A window size of k corresponds to training recalibration on forecasts within k weeks of the given forecast week, where available, inclusive. Log score is averaged over 9 seasons, 11 locations, and 29 weeks (higher log score is better). The largest window sizes slightly hurt the performance of the parametric model, and the smallest window sizes significantly hurt the nonparametric model. Averaged over all forecasters, the improvement in performance due to calibration is roughly equal to the improvement in performance by reducing the forecast horizon by a week.

2.3.3 Effect of number of training seasons

We chose to apply our recalibration to the FluSight Challenge because there are many forecasters available over many seasons for testing and training. When recalibrating forecasts of other epidemics, there may be significantly less training data available. Fortunately, these methods are robust to recalibrating FluSight Challenge forecasts with little training data. The parametric recalibration method improves the mean log score, averaged over all 27 forecasts, with just two training seasons, and the nonparametric recalibration improves average performance with four training seasons, as shown in Fig 2.8.

Because we train selectively based on seasonality, as discussed in Section 2.2.6, each training season and location contributes only 7 PIT values to estimate G. We pool 11 locations together, so the parametric method can improve performance with roughly 150 PIT values, and the nonparametric method can improve performance with roughly 300 PIT values.

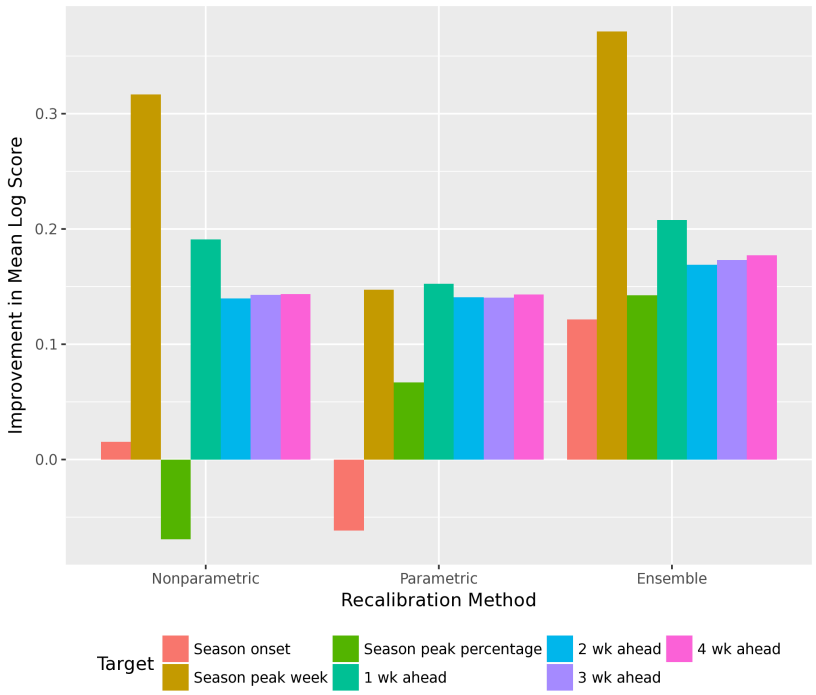

Figure 2.4. Improvement in mean log score, for the different recalibration methods. Log score is averaged over all 27 forecasters in the FluSight, 9 seasons, 11 locations, and 29 weeks (higher log score is better). The ensemble recalibration method improves accuracy for every target.

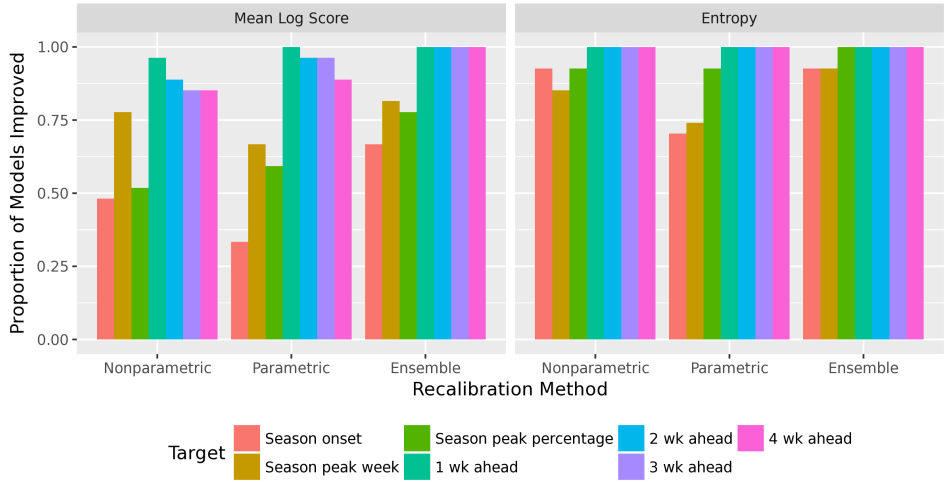

Figure 2.5. Proportion of forecasters for which recalibration improves mean log score (left) and entropy of the PIT values (right). The ensemble method improves accuracy for the short-term targets for all forecasters, and most forecasters for the seasonal targets. It also improves calibration (as measured by entropy) for most forecasters and most targets. The ensemble method outperforms both the nonparametric and parametric methods.

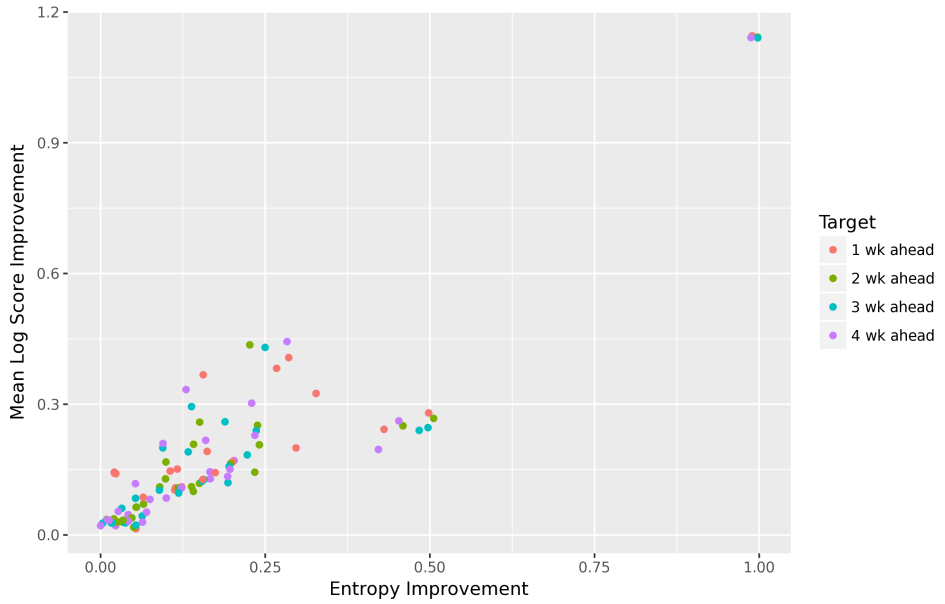

Figure 2.6. Improvement in mean log score versus improvement in entropy for each of the 27 FluSight forecasters and short-term targets. There is a clear linear trend (with slope approximately 1) between the improvement in calibration and the improvement in accuracy.

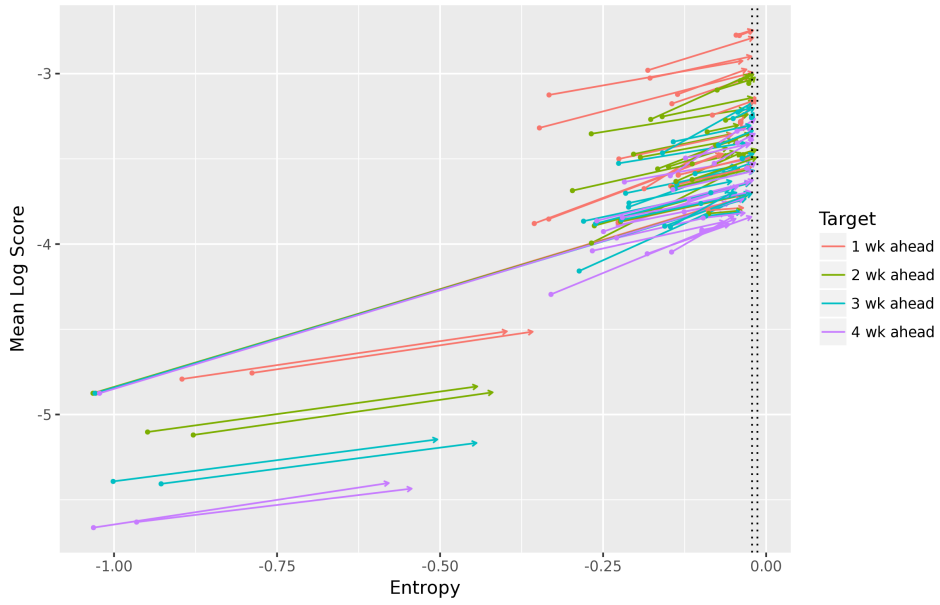

Figure 2.7. Entropy and mean log score before and after recalibration, for each of the 27 FluSight forecasters and short-term targets. The tail of arrow represents a quantity before recalibration, and the head after recalibration. The dotted lines show the central 90% interval of the entropy of a comparably-sized sample of standard uniform random variables for comparison. For all but two forecasters (the eight bottom-most line segments), the ensemble recalibration method achieves almost perfect calibration as evidenced by a near-zero PIT entropy, and this is accompanied by significant improvements in accuracy.

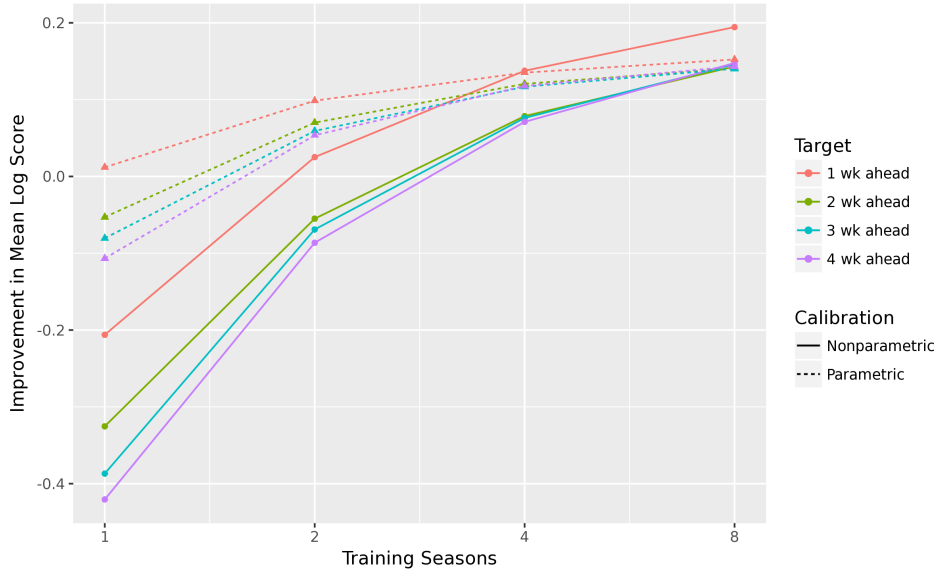

Figure 2.8. Improvement in mean log score after recalibration, averaged over all 27 FluSight forecasters, by number of training seasons. We perform three runs for each of the nine available seasons and $n \in \{1, 2, 4, 8\}$, where a run consists of randomly sampling n other seasons to train recalibration for each of the 27 FluSight forecasters. Each point in the plot is averaged over $9 \times 3 = 27$ runs. As expected, the parametric method is more robust to limited training data than the nonparametric method.

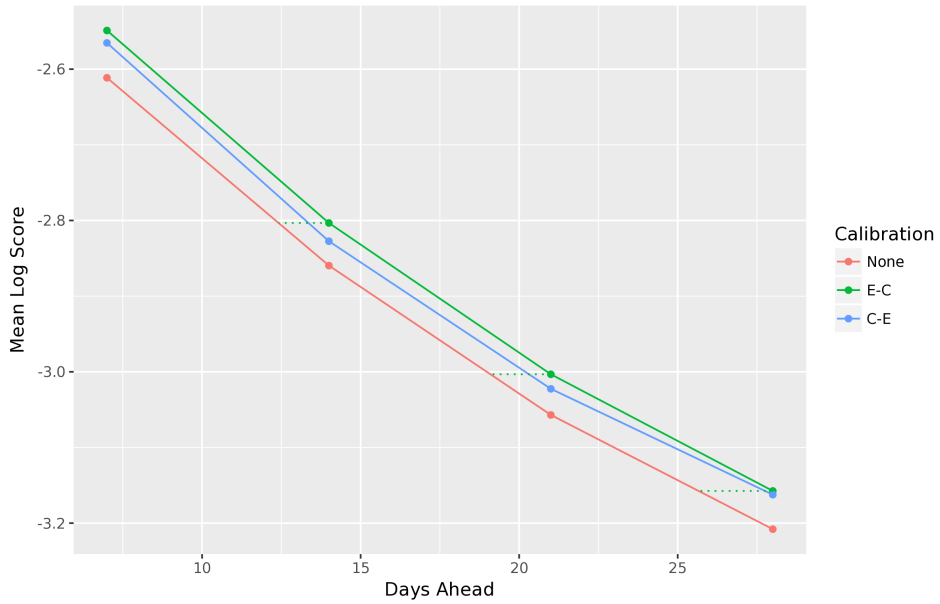

Figure 2.9. Mean log score for the two different approaches to recalibrating the FluSight ensemble forecaster, with C-E and E-C reflecting the order of recalibration and ensembling. Both the C-E and E-C models outperform the original ensemble (with no recalibration), but ensembling followed by recalibration performs best. By viewing forecast performance as a function of time, recalibration increases performance as much as roughly two days' time would.

2.3.4 Recalibrating the FluSight ensemble

As we just saw, recalibration improves the performance of the individual forecasters in the FluSight Network. A natural follow up is therefore to investigate whether it can improve the performance of the FluSight ensemble, a forecaster that combines 27 component forecasters (the individual FluSight forecasters), whose construction is described in [43].

As both recalibration and ensembling are post-processing methods (i.e., that can be applied in post-processing of forecast data), we are left with two options to explore. We can recalibrate the component forecasters and then ensemble (C-E), or ensemble the components and then recalibrate (E-C). In the C-E model, we train ensemble weights in a leave-one-season-out format, on the recalibrated component forecasts. In the E-C model, we train ensemble weights in a leave-one-season-out format on the original component forecasts, and then recalibrate the ensemble forecasts.

Fig 2.9 reveals that E-C model performs better than the C-E model. This is in line with established forecasting theory, which states that linear ensembles (which take a linear combination of component forecasters, such as the FluSight ensemble approach) themselves are generally miscalibrated, even when the individual component forecasters are themselves calibrated [28, 42, 20].

2.4 Discussion

Even in a domain as complex as epidemic forecasting, relatively simple recalibration methods such as those described in this paper can significantly improve both calibration and accuracy. A forecaster's performance for any proper score can be decomposed into three components: the inherent uncertainty of the target itself, the resolution of the forecaster (concentration of the forecasts), and the reliability of the forecaster to the target (calibration) [4]. In epidemic forecasting, without seasonality-aware recalibration training (such as that proposed and implemented in this paper), recalibration will not affect the resolution term, which is left to the individual forecasters, but it will improve the reliability term. However, using seasonality-aware recalibration, it can also improve the resolution term.

Over 9 seasons of forecast data from 27 forecasters in the FluSight Challenge, we found that recalibration was especially helpful for the short-term targets (1-4 week ahead forecasts). With the exception of two very similar forecasters that have poor performance, the ensemble recalibration method was able to reduce the entropy of the PIT distribution to nearly zero (not or barely statistically significantly different than a uniform distribution). The recalibrated forecasts are therefore more accurate and more reliable. This is true across a diverse set of forecasters, including mechanistic, statistical, baseline, and ensemble models; indeed, as our recalibration method treats the forecaster as a black box, it can be applied to any forecaster, given access to suitable training data (retrospective historical forecasts).

Recalibrating influenza forecasts avoids challenges present in other forecasting environments, such as nonseasonality, a lack of consistent forecasters spanning many seasons, and little training data. Although this makes recalibrating influenza forecasts a relatively easier task, we believe that this recalibration method can be applied to forecasting other diseases as well. For example, dengue fever is a seasonal disease with training data since 2014 available for forecasting [39]. Aedes mosquito counts are another seasonal target of interest to the CDC, which has released several years of training data for some counties for the purpose of forecasting [6]. This recalibration method, with its seasonal component, can be applied to these forecasts.

In application to nonseasonal diseases, such as COVID-19 (currently), this method can easily be modified to use all available PIT values, as opposed to the selective training used for influenza forecasts. Alternatively, selective training could be done not by calendar week but by some other feature(s) that differentiates a forecaster's behavior (e.g., whether cases are increasing or decreasing). While this allows for a flexible approach to recalibrate a variety of seasonal and nonseasonal diseases, this may be difficult to implement effectively in practice. In other cases where the PIT distribution changes slowly over time, training could be done only on the most recent forecasts to improve the estimate of $\hat{G}$. This selective training approach has been successful in recalibrating COVID-19 forecasts [40]. The ensemble approach allows for the incorporation of multiple models trained on different historical forecasts, or even different recalibration methods altogether.

Regarding a lack of consistent forecasters, even if a forecaster has been modified continuously over many years and previous performance is not indicative of current performance, recalibration can be trained on retrospective forecasts produced by the current forecaster.

A lack of training data is a more difficult problem to solve. An obvious problem of limited training data is the variance in estimating $\hat{G}$, but an additional challenge is that it is difficult to confirm our assumption that the PIT distribution is stationary over time. If we cannot detect that the PIT distribution changes over time, we will make inappropriate "corrections" to the forecasts that could harm calibration and accuracy. In practice, re-

calibration improved performance of the FluSight Challenge forecasts with relatively little training data, as shown in Fig 2.8. In less well-behaved applications, however, performance could decrease. We have made these recalibration methods available online so that a user can experiment with his or her own forecasts and determine whether or not recalibration improves performance [47].

The performance of recalibration with respect to the seasonal targets (onset, peak week, and peak percentage) was less conclusive than that of the short-term targets. Although the mean log score averaged over all of the forecasters was improved, recalibration only improved the performance of about three-quarters of the forecasters. Seasonal targets are inherently more difficult to recalibrate because at the end of the season, the true value has almost certainly been observed, and the forecasts are highly confident. For these forecasts, the correct bin has a mass of almost 1, and the observed PIT value then is approximately 0.5. At the end of the season, the PIT distribution is very concentrated at 0.5, which indicates underconfidence and poor calibration. If these PIT values of 0.5 are used to train forecasts for recalibration earlier in the season, before the target is observed, then recalibration incorrectly makes the forecast more confident. Because one is unsure whether the season peak has occurred or not for several weeks after the peak occurs, recalibration training is a nontrivial task. In general, more work is required to reliably improve accuracy and calibration for seasonal targets, which is a topic for future work.

2.5 Postscript: Calibration and Proper Scores

Above, we discussed the importance of calibration and demonstrated a method that both achieved calibration and improved forecasting accuracy. In Equation 2.1, we showed that recalibrating the forecasts by using the true PIT cdf G is guaranteed to improve the forecaster's expected log score. However, this property only holds for the log score; there are edge cases where achieving probabilistic calibration will actually lower performance when measured by another proper score.

Our goal is to connect recalibration methods and proper scores. Here, we focus on different CDF transform methods and show that there are certain CDF transform methods that are guaranteed to improve certain proper scores. We provide a framework for connecting a CDF transform method with its associated proper score guaranteed to be improved. Throughout, "guaranteed to improve" is shorthand for "guaranteed to have a nonnegative improvement".

2.5.1 Proper Scores

We begin by providing an overview of proper scores and their decomposition, summarizing relevant sections of [19] and [4].

There is a connection between proper scores and convex functions. Let $\mathcal{P}$ be the set of probability distributions over a sample space Ω. A scoring rule is a function $S : \mathcal{P} \times \Omega \to [-\infty, \infty]$. We also use $S(p, q)$ to denote the expected score of forecast p under q.

Scoring rule S is proper if and only if there exists a convex function G such that

$$S(p, \omega) = G(p) - \int G^*(p, \omega) dp(\omega) + G^*(p, \omega),$$

where G^* is a subgradient of G. The expected score $S(p, q) = G(p) - \langle G^*(p), p - q \rangle$.

$G(p)$ is equal to $S(p,p)$. It is a type of entropy function and it measures the best expected score achievable by a forecast where the true distribution is p. Note that here, higher scores are better. We do not wish to optimize G directly, that would merely give a p whose expected self-score is lowest (often the uniform distribution).

It is useful to know that G is convex because the divergence function $d(p,q) = S(q,q) - S(p,q)$ is a Bregman divergence under regularity conditions. Note that we follow Gneiting in reversing the usual order of the arguments of the Bregman divergence.

Now, consider a setup where a forecaster outputs a series of probabilistic forecasts p_i with corresponding true densities q_i[1]. The expected score is $\mathbb{E}S(p_i, q_i)$ where expectation is taken over i. This term can be decomposed into three components: uncertainty, resolution, and reliability.

$$\mathbb{E}S(p_i, q_i) = G(\mathbb{E}q) + \mathbb{E}d(\mathbb{E}q, q_i) - \mathbb{E}d(p_i, q_i),$$

where S, G, and d are defined as above and expectation is over i. The first term is the uncertainty term and measures how inherently difficult the forecasting task is. The second term is the resolution term and measures how different the truth is for different forecasts. The third term is the reliability, or calibration, term and measures how different the forecasts are from the truth.

Due to the assumption that each p_i is distinct, this decomposition is trivial. Optimizing the reliability term directly is equivalent to optimizing $\mathbb{E}S(p_i, q_i)$ because the other two terms are independent of p. The key takeaway from this decomposition is the connection between the third term and calibration.

2.5.2 Bregman Divergence

The Bregman divergence $d(p,q)$ for associated convex function G is defined as

$$d(p,q) = G(q) - G(p) - \langle \nabla G(p), q - p \rangle .$$

Here, $\nabla G(p)$ is itself a function from the sample space to $\mathbb{R}$. As $S(p,q)$ is defined as $G(p) - \langle \nabla G(p), p - q \rangle$, we have $d(p,q) = S(q,q) - S(p,q)$.

Score	$S(p,q)$	$G(p)$	$d(p,q)$	$\nabla G(p)(x)$
Log Score	$\int q(x) \log p(x)$	$\int p(x) \log p(x)$	$\int q(x) \log \frac{q(x)}{p(x)}$	$\log p(x) + 1$
Quadratic Score	$\int 2q(x)p(x) - \int p^2(x)$	$\int p^2(x)$	$\int (p(x) - q(x))^2$	$2p(x)$
CRPS	$-\int P^2(x) - 2P(x)Q(x) + Q(x)$	$\int P(x)(P(x) - 1)$	$\int (P(x) - Q(x))^2$	$\int_{y \geq x} (2P(y) - 1)$

[1] Technically, this formulation of the decomposition is true only when each p_i is distinct. Otherwise, the result is similar but mathematically more complex.

2.5.3 Convexity of Bregman Divergence

A Bregman divergence is convex in q but only in certain cases is it convex in p [2]. The Hessian of the Bregman divergence above with respect to p is

$$\frac{\partial^2 d(p,q)}{\partial p(x)\partial p(y)} = \frac{\partial^2 G(p)}{\partial p(x)\partial p(y)} - \int_z \frac{\partial^3 G(p)}{\partial p(x)\partial p(y)\partial p(z)} \cdot (q(z) - p(z))dz.$$

If we abuse inner product notation, we write the Hessian as

$$\nabla^2 G(p) - \left\langle \nabla^3 G(p), q - p \right\rangle.$$

When this Hessian is positive semidefinite, the Bregman divergence $d(p,q)$ is convex with respect to p. For the three scores above, the Bregman divergence is convex. By visual inspection of the table above, it is clear that the log score and quadratic score have a Bregman divergence whose Hessian is diagonal and always positive. Therefore, the Bregman divergence is convex.

The CRPS also has a convex Bregman divergence. The Hessian $\nabla^2 G(p)$ is a function: $(x,y) \mapsto \int_{z \geq x,y} 2dz$. So the term $\nabla^3 G(p) = 0$ and the Hessian of the Bregman divergence is convex because G is convex.

2.5.4 Defining Score-specific Calibration

The calibration term in the proper score decomposition is dependent on the choice of scoring rule. Thus it follows that the statement "This forecaster is calibrated" is also dependent on the choice of scoring rule or mode of calibration. We could define the term calibration to mean that the calibration term is 0, but this means that for all i, $p_i = q_i$. This is almost always unattainable, so this definition of calibration is not useful.

We propose defining calibration in the following way. Consider a post-processing method that applies a single CDF transform H applied to each P_i. That is, the new forecast $P_i^*(x) = H(P_i(x))$ and $p_i^*(x) = h(P_i(x)) \cdot p_i(x)$. A series of forecasts p_i is calibrated with respect to true densities q_i if

$$\arg\min_H \mathbb{E}d((h \circ P_i) \cdot p_i, q_i) = \text{identity}.$$

Note that this is equivalent to $h(\alpha) = 1$ for all α. Conceptually, this means that a CDF transform will not improve the calibration term in the proper score decomposition. Since the other terms are independent of H, it is also true that a CDF transform will not improve the expected score.

We can use the method of Lagrange multipliers to reparameterize this calibration condition.

$$\min_h \mathbb{E}d((h \circ P_i) \cdot p_i, q_i) \text{ such that } \int_\alpha h(\alpha)d\alpha = 1$$

$$\mathcal{L}(h,\lambda) = \mathbb{E}d((h \circ P_i) \cdot p_i, q_i) - \lambda \left(\int_\alpha h(\alpha)d\alpha - 1 \right)$$

Just for the optimization, let's define $f(h) = \mathbb{E}d((h \circ P_i) \cdot p_i, q_i)$ and $g(h) = \int_\alpha h(\alpha)d\alpha - 1$. We optimize the constrained problem when $\nabla f(h) = \lambda \nabla g(h)$. It is easy to see that $\lambda \nabla g(h)$ is a constant function that returns λ for any input of α. Therefore, we optimize the constrained problem when $\nabla f(h)$ is constant and has the same value for every α.

In general, the method of Lagrange multipliers will only identify stationary points. However, recall that $d(p, q)$ is a Bregman divergence. For scores where the Bregman divergence is convex in p, this method will find the global minimum. For this subset of proper scores (which includes the log score, quadratic score, and CRPS), we define a series of forecasts p_i as calibrated with respect to true densities q_i if

$$\frac{\partial \mathbb{E}d(p_i^*, q_i)}{\partial h(\alpha)} = \lambda \text{ for all } \alpha$$

when $h(\alpha) = 1$ for all α.

We present examples for the score-specific calibration condition for three of the most common proper scores: the log score, quadratic score, and CRPS.

Example: Log Score

$$\mathbb{E}d(p_i^*, q_i) = \frac{1}{N}\sum_i^N \int_x q_i(x) \log \frac{q(x)}{h(P_i(x))p_i(x)} dx$$

$$\begin{aligned}
\frac{\partial \mathbb{E}d(p_i^*, q_i)}{\partial h(\alpha)} &= \frac{1}{N}\sum_i^N \int_x q_i(x) \cdot \frac{p_i(x)\mathbb{I}[P_i(x) = \alpha]}{h(P_i(x))p_i(x)} dx \\
&= \frac{1}{N}\sum_i^N \int_\beta q_i(P_i^{-1}(\beta)) \cdot \frac{\mathbb{I}[\beta = \alpha]}{h(\beta)p_i(P_i^{-1}(\beta))} d\beta \\
&= \frac{1}{N}\sum_i^N \frac{q_i(P_i^{-1}(\alpha))}{h(\alpha)p_i(P_i^{-1}(\alpha))}
\end{aligned}$$

We see that a forecaster is log-score-calibrated when

$$\frac{1}{N}\sum_i^N \frac{q_i(P_i^{-1}(\alpha))}{p_i(P_i^{-1}(\alpha))} = 1 \quad \text{for all } \alpha,$$

where the constant value must be 1 because both p and q are densities. This is equivalent with the definition of probabilistic calibration in [18].

Example: Quadratic Score

$$\mathbb{E}d(p_i^*, q_i) = \frac{1}{N}\sum_i^N \int_x (h(P_i^{-1}(x))p(x) - q(x))^2 dx$$

$$\begin{aligned}
\frac{\partial \mathbb{E}d(p_i^*, q_i)}{\partial h(\alpha)} &= \frac{1}{N}\sum_i^N \int_x (2h(P_i(x))p_i^2(x) - 2p_i(x)q_i(x))\mathbb{I}[P_i(x) = \alpha] dx \\
&= \frac{2}{N}\sum_i^N \int_\beta (h(\beta)p_i(P_i^{-1}(\beta)) - q_i(P_i^{-1}(\beta)))\mathbb{I}[\beta = \alpha] d\beta \\
&= \frac{2}{N}\sum_i^N h(\alpha)p_i(P_i^{-1}(\alpha)) - q_i(P_i^{-1}(\alpha))
\end{aligned}$$

A forecaster is quadratic-calibrated when

$$\sum_i^N p(P_i^{-1}(\alpha)) - q(P_i^{-1}(\alpha)) = 0 \text{ for all } \alpha,$$

where the constant value must be 0 because both p and q are densities.

Example: CRPS

$$\mathbb{E}d(p_i^*, q_i) = \frac{1}{N}\sum_i^N \int_x (H(P(x)) - Q(x))^2 dx$$

$$\begin{aligned}\frac{\partial \mathbb{E}d(p_i^*, q_i)}{\partial h(\alpha)} &= \frac{1}{N}\sum_i^N \int_x (2H(P_i(x)) - 2Q(x))\mathbb{I}[P_i(x) \geq \alpha]dx \\ &= \frac{2}{N}\sum_i^N \int_{x \geq P_i^{-1}(\alpha)} (H(P_i(x)) - Q(x))dx\end{aligned}$$

This integral can only be the same for all α if the integrand is 0 everywhere. Therefore, a forecaster is CRPS-calibrated when

$$\sum_i^N (P_i(x) - Q_i(x)) = 0 \text{ for all } x.$$

This is equivalent to the definition of marginal calibration in [18].

2.5.5 Achieving Calibration with a CDF Transform

We can extend the results of the previous section to define H in such a way that we can apply H to a set of forecasts p_i and achieve calibrated forecasts p_i^*. We simply need to define H such that $\partial \mathbb{E}d(p_i^*, q_i)/\partial h(\alpha)$ is constant with respect to α. Once we have calculated the definition of calibration as above, it is straightforward to define H for a non-calibrated set of forecasts.

For example, for the log score, we would define

$$h(\alpha) = \frac{1}{N}\sum_i^N \frac{q_i(P_i^{-1}(\alpha))}{p_i(P_i^{-1}(\alpha))}.$$

For the quadratic score, we define

$$h(\alpha) = \frac{\sum_i^N q_i(P_i^{-1}(\alpha))}{\sum_i^N p_i(P_i^{-1}(\alpha))}.$$

The CRPS is trickier to recalibrate. We need to the following equation to hold for all x:

$$\sum_i^N (h(P_i(x))p_i(x) - q_i(x)) = 0.$$

This condition does not include α, so we cannot simply solve for $h(\alpha)$. The problem is similar to a system of equations, where h is the vector we want to compute, and the constraint above is an equation for each x. In matrix notation, we would want to solve $Ah = b$, where the variables are functions. We have

$$b(x) = \sum_i^N q_i(x) \text{ and } A(x, \alpha) = \sum_i^N p_i(x) \cdot \mathbb{I}[\alpha = P_i(x)].$$

This system does not have a closed form solution, but we can discretize h to be piecewise linear and select a set of x. We would solve this system of equations and get a piecewise linear h. However, the system is not even always solvable. Consider the following counterexample.

x	0	1
$P_1(x)$	0.1	0.5
$P_2(x)$	0.5	0.9
$Q_1(x) + Q_2(x)$	0.1	1.9

From the constraint at $x = 0$ (and that H is monotonically increasing from 0 to 1), we must have $H(0.5) \leq 0.1$, but from the constraint at $x = 1$, we must have $H(0.5) \geq 0.9$.

2.5.6 Summary

We noted that the classic proper score decomposition into uncertainty, resolution and reliability is somewhat degenerate in the case that all forecasts are unique. The first two terms are independent of the forecasts themselves, and the reliability/calibration term is equivalent to the overall score itself.

We provided a framework for connecting proper scores and calibration. For any proper score with a convex Bregman divergence, there is an associated definition of calibration which can be derived mathematically. This setup also allows for a score-specific recalibration method that takes an uncalibrated set of forecasts and calibrates them, and is guaranteed to improve performance for that score. We also demonstrated these steps using three common scoring rules.

Future work would account for the fact that in practice we do not have access to the full distributions q_i but only have a single sample drawn from each. To make this recalibration method practical, we will need to be able to estimate $h(\alpha)$ from observations. For the log score, the PIT values can be used to estimate h but a more general method will be needed for the general case.

Chapter 3

Extracting Signals from Insurance Claims Data

3.1 Introduction

Medical claims are an important component in disease surveillance. Each claim contains detailed information about an individual encounter with the healthcare system. Epidemiologists and statisticians can aggregate claims to quantify the burden of a given disease for a given location and time. Due to the detailed, multidimensional nature of a claim, the user must choose a method to aggregate and summarize all the claims in a given set to a measure representing the disease burden, often a single number. A common choice is the ratio of the number of visits for a set of conditions and the number of total all-cause visits, where the number of all-cause visits controls for differences in population size and dataset coverage. Because this ratio is dependent on the day of the week, the numerator and denominator are aggregated weekly in many studies of influenza [52, 9, 11, 34, 10, 32, 45, 1, 35]. Another common choice is to use the number of visits for a set of conditions and account for day-of-week effects by taking a moving seven day average. This allows for the signal to be available on the daily level.

The seven day average is a common choice, and removes artifacts in the signal due to the day-of-week effect. However, the data generating process of claims is often dependent on many factors other than just the day of the week. Healthcare-seeking behavior and capabilities also change on holidays and within a month. Claims data aggregated by an insurance provider is dependent on the population covered by that provider. Because insurance coverage usually begins on the first of a month, claims signals can experience discontinuities at month boundaries. Simple methods that estimate disease burden from claims data do not account for these holiday effects and therefore are biased. Ideally, a method that converts claims data into a signal of disease burden should correct for changes in health-seeking behavior.

Because disease burden changes smoothly over time, smoothness is a desired property of a signal as well. Applying a seven day average achieves this goal to some extent, but can result in discontinuities due to the boxcar kernel. Because each day of the week has a distinct behavior, the width of the kernel is fixed at seven days and cannot be adjusted to achieve more or less smoothing. A better signal would be more flexible, allowing a variable

level of smoothing and adjusting to the noise of the aggregated claims counts.

3.2 Methods

In this section, we describe the data and the process of extracting an influenza signal from the data.

3.2.1 Data

Optum, a subsidiary of United Health Group, provided the data for this work. We received counts of health insurance claims, aggregated by day, 5-digit ZIP code, and over six age groups. For each day, ZIP code, and age group, we received the count of all outpatient visits and the count of influenza-like outpatient visits. The influenza-like filter includes any claims with an ICD code in the range [J09* - J18*].

Our goal is to extract from this dataset a consistent measure of influenza activity, i.e. if we calculate the influenza level as ϕ on two separate days, then the influenza activity on both days is the same subject to unbiased noise. Two simple approaches are to count the number of influenza claims, or divide the number of influenza claims by the number of total claims. However, the data (health insurance claims) are generated in a very complex process that makes these two estimators severely biased.

The total number of influenza claims for a given location and day is dependent on the total number of people affiliated with insurance providers in the dataset. An individual may enter or exit the dataset when switching work to a company with a different insurance provider, or the workers of an entire company may enter or exit if the company changes insurance provider. The dataset itself is aggregated from several insurance providers, each of which may be represented only for a subset of the dataset's time range. For all of these reasons, the number of people covered in the dataset varies from month to month. It also varies between locations, as insurance providers have higher relative coverage in some areas than others. For these reasons, we cannot simply use the raw total influenza claims.

Additionally, health-seeking behaviors are very different on different days of the week and on holidays. This means we cannot simply use the ratio as an indicator of influenza activity. The number of both influenza and all-cause claims is far lower on weekends and holidays, but the ratio of influenza claims to total claims is higher on weekends and holidays. We suspect this is because routine and non-urgent visits are more often conveniently scheduled for a non-holiday weekday while acute visits are less likely to be delayed for a more convenient time. We also found increases in visits on the 1st and 16th day of the month. We are unaware of an underlying cause of these spike, and it is possible that this effect is simply a reporting artifact and unrelated to true healthcare behaviors. Therefore, we require a more complex model to generate a consistent measure of influenza activity.

We will model the two counts separately. First, we will use the total counts to estimate the number of people covered in the dataset for a given day (up to a constant). We will use that estimate to normalize the influenza counts. From the influenza counts and the normalization constant, we will control for temporary effects and extract a smooth influenza signal.

3.2.2 Denominator Model

We assume that on each day, the total number of outpatient visits is affected by the number of covered people in the dataset, a time-invariant propensity for each individual to see a doctor, and three temporary effects: day-of-week, holiday, and day-of-month. After inspecting the data, we assume the total number of people covered (up to a constant) θ_m changes only on month boundaries. This is reasonable because insurance coverage often becomes effective or ineffective on month boundaries. We also assume that each of the temporary effects is multiplicative. Therefore, our model is

$$\log \mu_t = \log \theta_{m(t)} + \alpha^*_{m(t),wd(t)} + \beta^*_{y(t),h(t)} + \gamma^*_{m(t),md(k)}, \tag{3.1}$$

where μ_t is the expected count of total outpatient visits on day t, $m(t)$ is the month of day t, $wd(t)$ is the day-of-week of day t, $h(t)$ is the index of the holiday of day t (if any), and $md(t)$ is the index of the special day-of-month of t (if either 1 or 16). The parameters α^* corrects for day-of-week effects, β^* corrects for holiday effects, and γ^* corrects for day-of-month effects (which only need to be corrected on the 1st and 16th of each month). We denote these parameters with a superscript star to emphasize that they are different from the temporal parameters in the following section. For identifiability, we constrain $\alpha^*_{m(t),\cdot}$ to sum to 0.

In this model, we do not assume that the holiday effects or day-of-month effects are similar across time. Therefore, we effectively ignore holidays and 1st and 16th of the month, because there is a free parameter for each of these days. We fit a model to maximize the Poisson log-likelihood to obtain θ, representing the total number of people covered (up to a constant). We retain θ_m for the model that describes influenza counts.

3.2.3 Numerator (Influenza) Model

To model influenza-like outpatient visits, we rely on similar assumptions as above. We assume that the number of visits on day t is dependent only on the number of covered people in the dataset ($\theta_{m(t)}$), the same three temporary effects as above, and additionally the influenza prevalence ϕ_t. This yields our model

$$\log \mu_t = \log \theta_{m(t)} + \alpha_{m(t),wd(t)} + \beta_{y(t),h(t)} + \gamma_{m(t),md(t)} + \log \phi_t, \tag{3.2}$$

where μ_t is the expected count of influenza-like outpatient visits, $m(t)$ is the month of day t, $y(t)$ is the year of day t, α corrects for day-of-week effects, β corrects for holiday effects, γ corrects for day-of-month effects, and ϕ_t is the influenza incidence on day t.

If we were to maximize solely the Poisson likelihood, the model would be non-identifiable, because there is a ϕ_t parameter for each day in the dataset in addition to all of the other parameters. Therefore, our objective function to minimize is

$$f(b) = -\ell(b; X, y) + \lambda_1 \|\Delta^3 \phi\|_1 + \lambda_2 \|\Delta_m \alpha\|_2^2 + \lambda_3 \|\Delta_y \alpha\|_2^2, \tag{3.3}$$

where $\ell(b, X, y)$ is the Poisson likelihood of the parameters given the data, $\Delta\phi$ is the daily change in ϕ_t, $\Delta_m\alpha$ is the monthly change in α for a given day of the week, and $\Delta_y\alpha$ is the yearly change in α for a given day of the week. We impose these penalties because we assume 1) influenza prevalence changes smoothly over time but has curvature, 2) day-of-week effects change smoothly over time, and 3) day-of-week behaviors are seasonal and change smoothly from year to year.

3.2.4 Model Evaluation

A standard cross validation approach to select hyperparameters does not work in our case because we lack ground truth influenza prevalance data. However, we can do cross validation by assuming that influenza prevalence is nearly linear on a scale of three days. We train the model on two thirds of the data, leaving out every third day. The model returns α for all of the months and ϕ for two of every three days. For a left-out day t, we can get the model's estimate of the influenza level at day t as the arithmetic average of the influenza level on the neighboring days $\hat{\phi}_t = (\phi_{t-1} + \phi_{t+1})/2$.

For each left-out day t, we calculate the Poisson likelihood of seeing a count y_t with a mean

$$\log \hat{\mu}_t = \theta_{m(t)} + \alpha_{m(t),wd(t)} + \beta_{y(t),h(t)} + \gamma_{m(t),md(t)} + \log \hat{\phi}_t$$

We do 3-fold cross validation and calculate the average Poisson likelihood for each set of hyperparameters $\{\lambda_1, \lambda_2, \lambda_3\}$.

Alternatively, we could evaluate models in an in-sample fashion, by training the model on all of the data and still estimate $\hat{\phi}_t$ as the average of ϕ_{t-1} and ϕ_{t+1}. This likely leads to overfitting by smoothing the ϕ curve too strongly.

3.2.5 Two signals: ϕ and ξ

Our method allows one to extract two different signals with different interpretations from the aggregated claims counts. The first signal, ϕ, is fit directly by optimizing the objective in 3.3. The second signal is ξ, and is can be calculated from the other parameters fit in 3.3. We define our two signals ϕ_t and ξ_t as

$$\begin{aligned} \log \phi_t &= \log \mu_t - \theta_{m(t)} - \alpha_{m(t),wd(t)} - \beta_{y(t),h(t)} - \gamma_{m(t),md(t)} \\ \log \xi_t &= \log y_t - \theta_{m(t)} - \alpha_{m(t),wd(t)} - \beta_{y(t),h(t)} - \gamma_{m(t),md(t)} \end{aligned} \tag{3.4}$$

Both signals represent the underlying disease burden, after correcting for temporal effects. However, ϕ is calculated from the latent variable μ, whereas ξ is calculated from the observed counts y. Because of the third-difference penalty in 3.3, ϕ is smoothed temporally but ξ is not. In many applications, smoothness is a desired property of a latent signal, so ϕ will be preferable. However, temporal smoothing can blur a peak or a trough, so in some instances, ξ will be preferable to achieve higher resolution of the disease signal.

In locations with small counts, ξ may be too noisy to be meaningful without smoothing. This can be accomplished with trend filtering, which yields a result conceptually similar to ϕ, or by taking a moving average, which yields a result conceptually similar to the seven-day average.

3.3 Results

We present a comparison of four signals that can be derived from the insurance claims dataset: ϕ, ξ, $\xi - 7dav$, and $7dav$. The signals ϕ and ξ were introduced in Equations 3.2 and 3.4 respectively, where both signals apply bias correction, but ϕ is optimized to be smooth and ξ is not. We take a moving seven day average of ξ to obtain $\xi - 7dav$, and $7dav$ is a simple ratio of the moving seven day average of influenza visits divided by the moving seven day average of the total outpatient visits. Here, $7dav$ serves as a baseline, as it is a standard option in deriving a signal from insurance claims data.

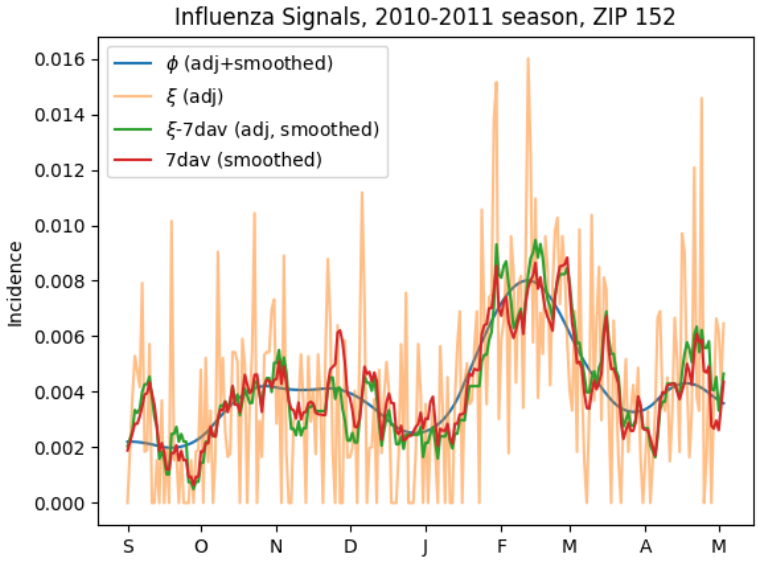

Figure 3.1. Four signals derived from our insurance claims dataset, for the 2010-2011 influenza season in 3-digit ZIP 152 (Pittsburgh).

We provide an example visualization of these four signals in Figure 3.1. As would be expected, ϕ is the smoothest of the signals, due to the direct penalization of the third differences. The signal with the most noise is ξ, as this signal has no temporal smoothing at all. The other two signals, $\xi - 7dav$ and $7dav$, track each other very closely. Although they are calculated quite differently, both are calculated directly from the influenza counts and use a seven day moving average. The main practical difference is that $\xi - 7dav$ accounts for holiday effects while $7dav$ does not. We see this in Figure 3.1, where $7dav$ is significantly higher during the holiday periods in late November and late December.

3.3.1 Influenza Analysis

In this section, we present results of basic analyses of influenza dynamics, using each of the four signals described above. The motivation for these analyses is drawn from several studies of influenza cited in the introduction. We will take peak timing as our main metric of a season's timing within a location, because it is simple conceptually and has an obvious mathematical definition, namely the day at which the influenza signal is highest.

We plot peak timing in Figure 3.2. We see a bimodal distribution for all four signals, with one mode in late December and another in late February. In the six week period between mid-December and the end of February, there are approximately the same number of total peak timings for each of the four signals, but for the $7dav$ signal, they are almost all concentrated in the last week of December and the first week of January. This corresponds exactly to the timing of Christmas and New Year's Day, two of the biggest holidays in the United States. Given that we expect the proportion of influenza visits to spike on holidays, it is certainly plausible that in many cases, the true influenza activity did not actually spike

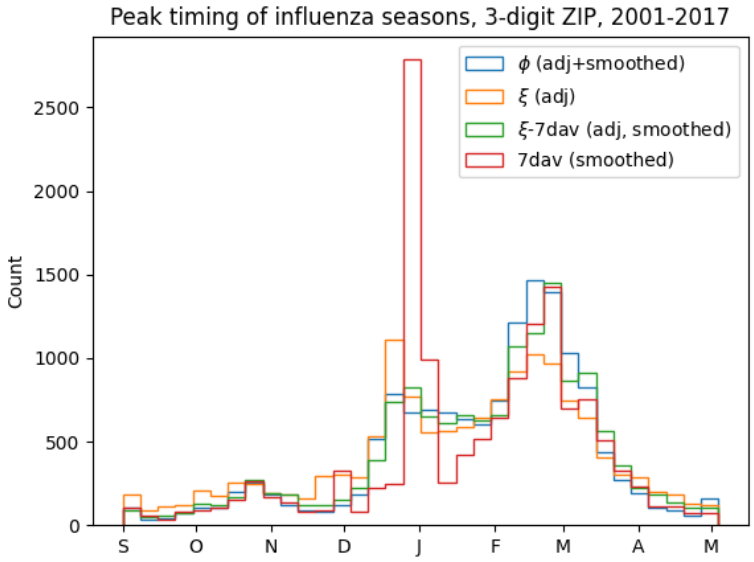

Figure 3.2. For each location and season, a histogram of the week in which peak influenza intensity occurred, for four signals. The histogram is similar for all of the signals, with the exception of a six week period around the turn of the calendar year. The last week of December and first week of January see a large spike in peak timing for the $7dav$ signal, but not for the other three. During the two weeks before and after this period, the other signals have higher occurrences of peak incidence than the $7dav$ signal.

during those weeks. Rather, the true peak occurred a week before or after, but the holiday spike caused the $7dav$ peak to be measured during the holiday. The temporal corrections performed by the other signals avoid these spurious peaks.

We can use differences in peak timing to measure the synchrony between two locations. We will quantify synchrony as the mean absolute difference between peak timing in two locations. The closer the peak timing is in two locations, the more synchronous they are. In Figure 3.3, we measure pairwise synchrony as a function of distance. All four signals show the same trend: synchrony decreases rapidly as distance increases to 200 miles, then decreases slowly until distance increases to 1500 miles, then flattens. The main difference between the signals is the magnitude of the mean absolute difference of peak times, which is likely due to the noise present in the signals. The signal with the lowest difference is ϕ, which has the least noise. The $7dav$ signal has a smaller average difference than $\xi - 7dav$, possibly because the $7dav$ signal sees many peaks during the year-end holiday season. The signal ξ has the largest difference, and is the noisiest. Despite the differences in the peak timing histogram, synchrony is similar across all signals.

3.3.2 Validation

The absence of ground truth creates a significant challenge in validating our approach. We have derived three signals from our dataset that we believe should better model the true

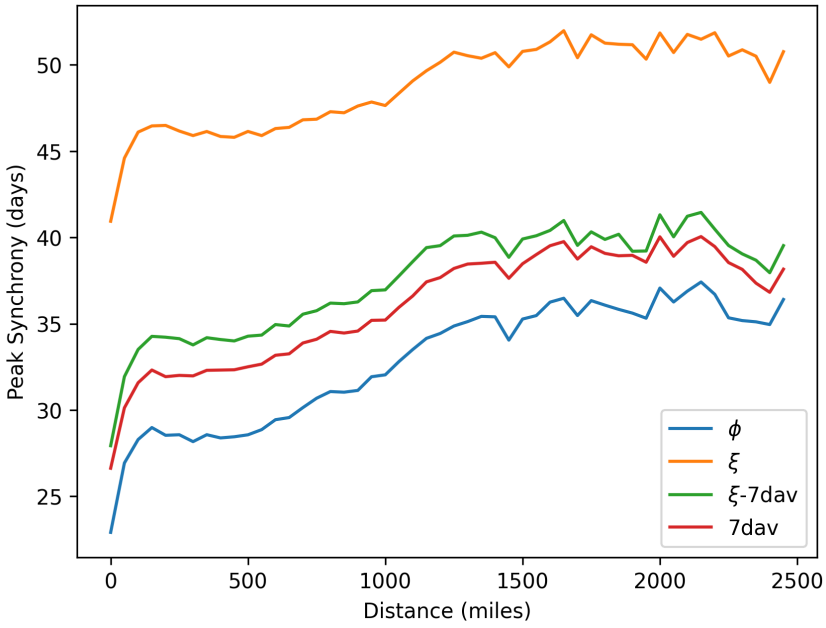

Figure 3.3. We plot synchrony in peak timing as a function of distance between two locations. For each pair of locations, we calculate the mean absolute difference in peak timings over all 16 seasons to quantify synchrony in peak timing. We then bin pairs of locations based on geographical distance and average the peak timing synchrony. For all four signals, the overall pattern is the same, with synchrony decreasing as a function of distance sharply until 200 miles, then slowly until 1500 miles, then flattening. The actual difference in peak times varies widely by signal, this is likely due to the noise present.

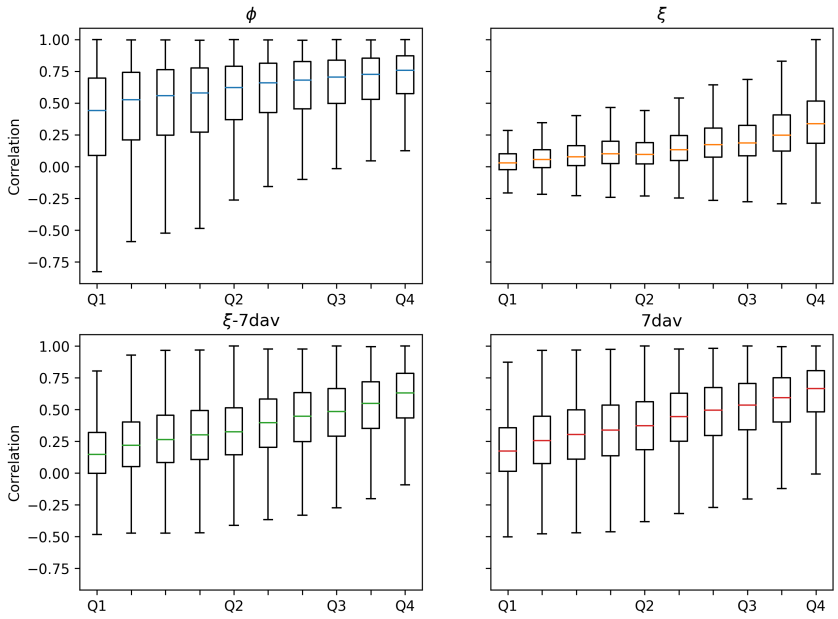

Figure 3.4. We plot correlation of influenza activity as a function of the product of population sizes for each pair of locations. Locations are divided into four sets by quartiles of population size (Q1 is lowest). For all signals, correlations increase as population size increases.

influenza incidence than the naive seven day average signal. In a conventional setup, we would validate that our model outperforms the baseline by comparing the error between our signals and the baseline. Because we lack ground truth, we will evaluate the different models by how well they predict the influenza counts in the original dataset itself.

We assume that a signal which measures the true influenza incidence is easier to forecast than a signal which is affected by temporal biases. A simple experiment would be to train a forecasting model to predict each of our signals and report the signal with the smallest forecasting error. However, a smoothed signal is inherently easy to forecast, and a degenerate signal which has a value of 0 everywhere is the easiest to forecast. To control for smoothing, we train a forecasting model to predict the signal itself, but then convert the signal to counts and calculate the forecasting error relative to the original influenza counts in the dataset.

Our validation setup is as follows:

1. Fit ϕ, ξ, $\xi - 7dav$, and 7dav for each location, on all time points.
2. For each signal, create matrix $X \in \mathbb{R}^{16\times 35}$ for training and testing, where $X_{i,t}$ is the value of the given signal in season i and week t (i.e. $7 \cdot t$ days after September 1 of season i).
3. For each signal and season s, evaluate the model:
   (a) Create training matrix X^s_{tr} by removing row s from X and then removing seasonality by subtracting the column means.
   (b) Fit an autoregressive model of order 3 (AR-3) to the data in X^s_{tr}.

(c) Predict $\hat{X}_{s,3:35}$ using the AR-3 model fit in the previous step.

(d) Convert the predicted signal $\hat{X}$ to the predicted influenza count mean $\hat{Y}$.

- For ϕ, ξ, and $\xi - 7dav$, estimate the day-of-week parameters α by averaging the previous month's and next month's value. Use the fitted values for holiday (β) and day-of-month (γ) parameters, which are constant across the entire dataset. Add the appropriate temporal effects and covered population (θ) using equation 3.2 or 3.4.
- For 7dav, multiply the signal by the observed seven day sum of total outpatient to obtain the estimated sum count of influenza visits over the last seven days. Then multiply by the dataset-wide proportion of influenza counts observed on the given weekday in order to get the prediction mean.

(e) For each signal, assume the predicted distribution of influenza counts is a Poisson with the mean specified in the previous step. Then calculate the likelihood of the observed count.

In this setup, we create a simple AR-3 predictive model to forecast the signal values within a season. In Step 3d, we need to convert the signal to the influenza count scale differently depending on the signal. For ϕ and ξ, we accomplish this by directly using the α, β, γ, and θ parameters fit by our model. For the seven day average, we multiply the estimated influenza fraction by the raw total outpatient counts to obtain the estimated count of influenza visits. To provide a fair comparison, all predictions are made on the scale of counts of a single day.

We note that we would be unable to produce the predictions of the counts themselves in real-time, as they rely either on the temporal effects or observed total counts for the prediction date itself. However, to validate by producing real-time influenza count predictions would require predicting other parameters. For ϕ and ξ, we would have to predict α and θ, and for 7dav, we would have to predict the total outpatient visits. This introduces confounding in our comparisons, as one of those quantities may be easier to predict than the other. Therefore, we focus on predicting only the signal, and convert to the count scale by using data unavailable in real time.

We show the results of our validation experiment in Figure 3.5. We see that ϕ has the highest likelihood. The signals $\xi - 7dav$ and $7dav$ perform similarly, with some exceptions, and the ξ signal has the lowest likelihood. In Figure 3.1, we saw that $\xi - 7dav$ and $7dav$ had similar values except during holiday periods. The validation likelihood is also similar except during holiday periods. During the holiday periods, the likelihood drops substantially for the $7dav$ signal relative to the $\xi - 7dav$ signal. The unsmoothed ξ signal, which is by far the noisiest, has the lowest likelihood overall, but still performs better than $7dav$ during the holiday periods.

These results indicate that the temporal corrections produce signals which better represent the patterns within the insurance claims dataset than the naive $7dav$ signal. The holiday corrections are not merely extra parameters that allow for overfitting, but are necessary in removing biases. Given that ϕ outperforms $\xi - 7dav$, we conclude that our model's smoothing is more effective than seven-day average smoothing.

3.3.3 Sensitivity Analysis: Naive Smoothers

As mentioned above, the seven day average is a common approach to avoid the day-of-week biases in the data. However, the seven day average can still be quite noisy, and a longer

Figure 3.5. We show the mean Poisson likelihood of the observed influenza counts for each signal. The ϕ signal performs best in our validation experiment, followed by $\xi - 7dav$, and then $7dav$. The unsmoothed ξ signal performs worst.

smoothing window may provide a better signal by optimizing the bias-variance tradeoff. We show below in Figure 3.6 that a longer smoothing window can perform almost as well as our smoothed and corrected signal ϕ on the validation setup from the previous section. During normal periods of the year, a 14 or 21 day averaged signal can perform just as well or better as the ϕ signal. During the holiday periods, however, the naive averaged signals perform significantly worse than the ϕ signal. Overall, this indicates that the method of smoothing (simple n-day averaging or more complex local quadratic smoothing) may not make a significant difference in the accuracy of the signal. However, correcting for holiday effects is important to maintain an accurate signal.

3.3.4 Ablation Study

We performed ablation experiments to determine the importance of each of the model's components. We analyze three ablation models, each removing a different component of either Equation 3.2 or Equation 3.3: holiday effects (β), day-of-month effects (γ), and day-of-week penalties (λ_2 and λ_3). We then fit ϕ values for each of these models and compare their validation accuracy, using the validation setup above. As shown in Figure 3.7, each of these models performs worse than the original model, indicating that all of these components are necessary in removing biases.

As expected, removing the corrections for holiday effects decreases validation accuracy most during the holiday periods of Thanksgiving and Christmas/New Year's. However, the holiday-ablation model performs worse throughout the rest of the season as well. We identify two potential causes. The first is that the day-of-week corrections α are biased due to the holidays falling on a given day of the week. Due to the month-to-month penalties

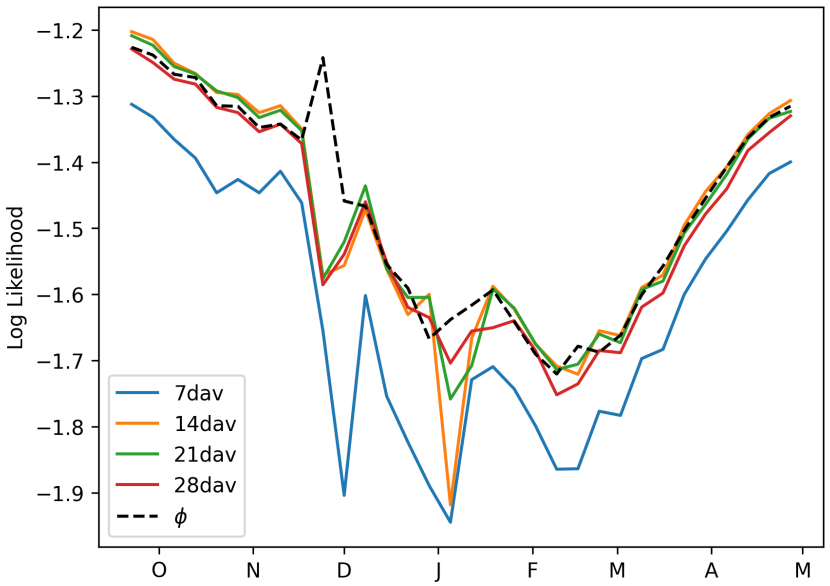

Figure 3.6. We show the mean Poisson likelihood of the observed counts for different simple averaged signals. The 14-day average performs the best, and slightly outperforms ϕ during non-holiday periods. However, it performs very poorly during holiday periods and is overall less accurate than ϕ.

(λ_2) in the objective function 3.3, a strong bias in the α values for one month can cause biased α values in neighboring months. A second potential cause is the smoothness penalty (λ_1) on ϕ in Equation 3.3. Ensuring local smoothness can cause biased ϕ values during the holiday period can propagate to neighboring time periods as well.

Removing the corrections for the day-of-month effects also reduces validation accuracy, although to a lesser extent than removing holiday effects. This is also expected, as the day-of-month effects are less pronounced in the dataset. Similar to the holiday-ablation model, accuracy is reduced throughout the entire season. Both the λ_1 and λ_2 penalties are potential causes here as well.

The α-penalty-ablation model performs the worst by far. The reason is simple: with only four or five data points per fitted α value, the day-of-week corrections overfit to the data in the absence of regularization. In the validation experiment, we estimate the value of α for a month by interpolation between the previous and next months' values. In the presence of overfitting, the interpolated day-of-week effects are highly variable and inaccurate. Additionally, overfitting the day-of-week corrections results in values of ϕ that are also potentially inaccurate.

Overall, the ablation experiments show that each of the model's components is necessary to identify and correct for temporal biases in the data.

3.4 Discussion

Health insurance claims have tremendous potential as a data source for infectious disease surveillance. Their biggest strengths are the size of the population covered in the dataset

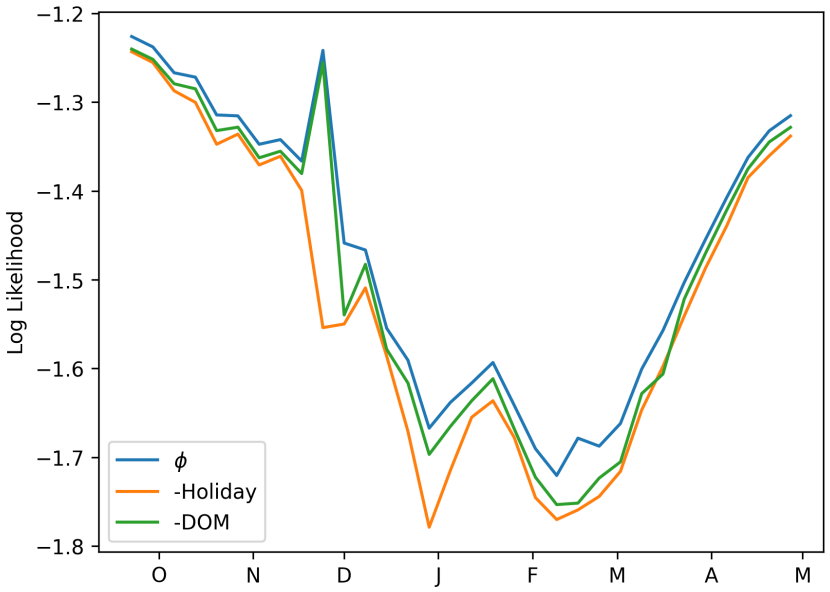

Figure 3.7. We show the mean Poisson likelihood of the observed counts for two of the ablation models. In each case, removing a component of the model reduces accuracy. Removing day-of-month (DOM) effects decreases accuracy throughout the entire month, and removing holiday effects decreases accuracy during the holiday period. The model without the λ_2 and λ_3 penalty terms is not shown here, as it is far less accurate and obscures the differences between the other three models (average log likelihood = -2.81.

and the fine spatial and temporal granularity. However, biases in the data due to changes in healthcare-seeking behaviors mean that the data cannot simply be used as-is. Through a combination of domain knowledge and data analysis, we can find the underlying factors which distort the claims signal and correct for them.

In the absence of ground truth, it is difficult to confidently validate that we truly remove temporal biases. By assuming that temporal biases impair predictive accuracy, we created an validation setup to quantify how closely a signal measures the underlying flu incidence present in the dataset. The two seven day average signals performed very similarly, with the exception of two holiday seasons: Thanksgiving and Christmas/New Year's. During those two periods, $\xi - 7dav$ which corrects for holiday effects performed much better than $7dav$, the naive seven-day average signal. Our inherently smoothed signal ϕ performed even better, which suggests that a locally quadratic smoother is more appropriate than a seven-day average, which effectively uses a sliding boxcar kernel.

Once we have an unbiased signal, we can perform detailed analyses and be more confident that the output is meaningful. For example, relatively simple summaries such as timing and intensity of influenza seasons are more accurate and can be estimated with higher precision. More complex metrics are more reliable when calculated from an unbiased signal.

A limitation of this method is that it requires a priori knowledge of the bias-inducing factors, whether through domain knowledge or data analysis. These factors are likely common to all insurance claims datasets, but this method does not detect the biases which it corrects. Therefore, it may need significant adaptation to be applied to other types of datasets.

Another limitation is that the method cannot be used in real time. Several of the parameters are unique to a given month, and without at least two weeks of data, the fitted coefficient values will be highly variable. This currently prevents the method from being used as a covariate for real-time forecasting or a target for forecasting. Future work could extend our method to allow real-time utility of the bias-corrected insurance claims signal.

Chapter 4

Correcting for Spatial and Temporal Heterogeneity

4.1 Introduction

Understanding the burden of epidemics is a critical task for both public health officials and modelers. However, traditional surveillance signals are often not available in real-time, due to delays in data collection as well as data revisions. Alternative data sources can provide more timely information about an epidemic's current state, which can be useful for modeling and forecasting. We can use these data sources to create *indicators*, which provide a single number quantifying some measure of epidemic burden for a given location and time. An indicator usually estimates the disease burden at a certain severity level (e.g. symptomatic infections, hospitalizations) when the ground truth is unobserved. During the COVID-19 pandemic, the Delphi group published a repository of several real-time indicators of COVID-19 activity [44].

Many, if not all, of these indicators suffer from heterogeneity. That is, the relationship between the indicator and unobserved ground truth changes over space or time. To define heterogeneity, let $X \in \mathbb{R}^{N \times T}$ be the matrix containing the indicator values for N locations and T time values, and $Z \in \mathbb{R}^{N \times T}$ be the matrix containing the corresponding ground truth values. We say that spatial heterogeneity is present when

$$\mathbb{E}[X_{i_1 t}] - Z_{i_1 t} \neq \mathbb{E}[X_{i_2 t}] - Z_{i_2 t} \text{ for some } i_1 \neq i_2, t.$$

Likewise, temporal heterogeneity is present when

$$\mathbb{E}[X_{i t_1}] - Z_{i t_1} \neq \mathbb{E}[X_{i t_2}] - Z_{i t_2} \text{ for some } i, t_1 \neq t_2.$$

Note that we define heterogeneity not simply as a bias in the indicator, but rather that the bias is dependent on location or time. The causes of heterogeneity vary depending on the indicator, but we can consider as an example an indicator based on insurance claims that seeks to estimate incidence of COVID-19 outpatient visits. Insurance claims could be higher relative to COVID-19 incidence in locations where the population in the insurance dataset is older, or where the doctors have more liberal coding policies in labeling a probable COVID case. Even the signal of reported cases, which purportedly reflects COVID-19 infections directly, will suffer from heterogeneity. If a few locations suffer from a shortage of tests, or

from a new strain which tests are less accurate in detecting or that has a different fraction of symptomatic cases, those locations will have a different relationship between reported cases and true cases. Similar causes can result in temporal heterogeneity. Test shortages, changing demographics, coding practices can also vary over time within a single location. For example, spatial heterogeneity has been documented in CDC's ILINet due to different mixtures of reporting healthcare provider types in the network [29].

We use real-time indicators for three main functions: modeling the past, mapping the present, and forecasting the future. Correcting for heterogeneity is important for all of these applications. Any statistical conclusions we make about spatiotemporal spread of a disease may be distorted if the underlying data is subject to heterogeneity. In the presence of spatial heterogeneity, the indicator values are not comparable across locations, and a choropleth map displaying the current values of the indicator will be misleading. Similarly, in the presence of temporal heterogeneity, displaying a time series of the indicator may be misleading. Heterogeneity affects forecasts as well, as biases in the features of a forecasting model will lead to forecast inaccuracy. Our goal is to remove heterogeneity in an indicator in order to make it more reliable for these three uses.

Heterogeneity has been described and modeled in the field of econometrics [50]. Nearly all of the work involving heterogeneity in econometrics deals with the implications in regression. If only spatial heterogeneity is present, then a fixed or random effects model can be used [23, 53]. Others have developed parametric methods that assume heterogeneity is also time-varying [33]. The main reason that these methods cannot be transferred to our domain is that they identify heterogeneity only through strict assumptions on the error terms in the regression model. Additionally, we are not performing regression in our application. Rather, we are trying to remove the heterogeneity in the indicator.

A challenge of correcting for heterogeneity is that the problem doesn't have a clear formulation. In nearly every practical application, we lack access to the ground truth and our best option is to compare our indicator with another signal that is a noisy estimate of the ground truth, and often suffers from heterogeneity itself. We will call this signal a "guide" to emphasize that it is not a target for prediction. We believe that the indicator is strongly related with the guide, so they should be correlated across time and space. However, they don't measure the same value, so the correlation should not be 1 even in the absence of noise. Another challenge is that we present the problem in a retrospective setting, without a clear division for training and testing.

In this paper, we investigate removing heterogeneity from two indicators using a different guide for each. The first indicator is based on insurance claims data, and we use reported cases as a guide signal. The second indicator is based on Google search trends of specific queries related to COVID-19. We use the COVID-19 Trends and Impact Surveys (CTIS) as a guide. All of these signals (indicators and guides) are available in COVIDCast [44].

Because heterogeneity is present in a wide variety of indicators, we desire a solution that is general and flexible. Another desired property is that the temporal corrections are smooth across time, because we want to accommodate situations where the relationship between the indicator and guide can drift slowly over time. The model should be flexible enough to allow for abrupt changes, but these should be limited in number. If the corrections are jagged in time, the model may be overadjusting to the guide signal rather than identifying and removing the true heterogeneity.

Lastly, the method should generalize well to a variety of indicators and guides. It should not rely on specific domain knowledge of a single indicator-guide pair because we want the method to be applicable to any current or future indicator and guide. If we believe the

indicator and guide have a stronger relationship, then we might want the model to use the guide matrix more and make a stronger bias correction. If we believe that there is more noise in the guide variable, that heterogeneity is mild, or that the inherent signals are more divergent, we might want the model to make a weaker bias correction. Additionally, the temporal smoothing constraint will be stronger or weaker, depending on the application.

The model should have hyperparameters to control the strength of the guide signal in fitting as well as the strength of the temporal smoothness constraint. These can be conceptualized as "knobs". For the indicator-guide relationship, the knob turns between one extreme of not using the guide signal at all and the other extreme of fitting maximally to the guide signal (in some models, fitting exactly to the guide signal). For the temporal smoothness constraint, the knob turns between the extremes of applying no smoothing and enforcing a constant temporal correction factor across time.

In the rest of this paper, we will provide three methods to correct for heterogeneity for a general indicator and guide signal. We then demonstrate their performance in simulated experiments and on several actual epidemiological data sources.

4.2 Methods

Let $X \in \mathbb{R}^{N \times T}$ be the matrix containing the indicator values for N locations and T time points, and $Y \in \mathbb{R}^{N \times T}$ be the matrix containing the corresponding guide values. We want to transform X to a matrix $\tilde{X}$, with the spatial and temporal biases mitigated. As mentioned above, the simplest way to do so is to set $\tilde{X} = Y$, but this is the most extreme version of overadjustment and removes any unique information contained in X. We will present three methods to remove heterogeneity by using Y as a guide. The first uses a simple low-rank approximation, and the second and third add elements which ensure that the biases removed are smooth in time. In all of our methods, we detect heterogeneity by examining the difference $Y - X$. We assume that the signal in this difference matrix is the heterogeneity between X and Y.

4.2.1 Bounded Rank Approach

In this approach, we assume that the heterogeneity between X and Y is of low rank. We begin without making any assumptions on the smoothness of the temporal biases. Therefore, we solve the following optimization:

$$\hat{A}, \hat{B} = \arg\min_{A,B} \|(X + AB^T) - Y\|_F^2,$$

where $A \in \mathbb{R}^{N \times K}$, $B \in \mathbb{R}^{T \times K}$, $K \leq \min(N, T)$, and $\| \cdot \|_F$ is the Frobenius norm. This optimization can be solved by performing singular value decomposition on the difference matrix $Y - X$ and keeping the vectors with the K highest singular values. The corrected matrix is $\tilde{X} = X + AB^T$.

4.2.2 Fused Lasso Approach

In addition to the low rank assumption, here we further assume that the temporal biases are mostly piecewise constant over time. Therefore, we solve the following optimization:

$$\hat{A}, \hat{B} = \arg\min_{A,B} \|(X + AB^T) - Y\|_F^2 + \lambda \|\Delta_t B\|_1,$$

where $A \in \mathbb{R}^{N \times K}$, $B \in \mathbb{R}^{T \times K}$, and $K \leq \min(N, T)$, and $\Delta_t B$ contains the first differences of B along the time axis. The $\Delta_t B$ penalty is inspired by the fused lasso [48] and encourages B to be piecewise constant along the time axis.

We solve this optimization using penalized matrix decomposition algorithms described in [55]. We reproduce the algorithm as applicable to our case here:

1. Let $Z^1 = Y - X$.
2. For $k = 1, \ldots, K$:
   (a) Initialize v_k to have L_2 norm 1.
   (b) Iterate until convergence:
      i. If $v_k = 0$, then $u_k = 0$. Otherwise, let $u_k = \frac{Z^k v_k}{\|Z^k v_k\|_2}$.
      ii. Let v_k be the solution to

$$\min_v \frac{1}{2}\|Z^{kT} u_k - v\|_2^2 + \lambda \sum_{j=2}^{T} \|v_j - v_{j-1}\|_1.$$

   (c) Let $d_k = u_k^T Z^k v_k$.
   (d) Let $Z^{k+1} = Z^k - d_k u_k v_k^T$.
3. A is the matrix whose k^{th} column is $d_k u_k$, and B is the matrix whose k^{th} column is v_k.

Step 2b) ii) is a fused lasso problem and can be solved using the alternating direction method of multipliers (ADMM) [3]. All of the other steps are trivial to compute.

This optimization has two hyperparameters which can be considered as "knobs": K and λ. The matrix rank K controls the degree to which we match the guiding signal Y. When $K = 0$, we keep X exactly as is and apply no correction. As K increases, we use more information from Y, and when $K = \min(N, T)$, we transform X to equal Y exactly (when $\lambda = 0$). The lasso penalty λ enforces smoothness along the time axis of B. At $\lambda = 0$, we apply no smoothing at all, and the model is equivalent to the Bounded Rank Model above. As λ approaches ∞, B contains a constant value across each row.

4.2.3 Basis Spline Approach

An alternative way to enforce smoothness on the temporal bias correction is to transform the temporal corrections by using B-spline basis functions. These functions S are determined by setting the polynomial degree d and a set of knots $\{t_1, \ldots, t_m\}$ [14]:

$$S_{i,0}(x) = 1, \text{ if } t_i \leq x < t_{i+1}, \text{ otherwise } 0,$$

$$S_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} S_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} S_{i+1,k-1}(x),$$

for $i \in \{1, \ldots, m\}$ and $k \in \{1, \ldots d\}$. We can use these basis functions to create a fixed spline transformation matrix $C \in \mathbb{R}^{L \times T}$, where $C_{i,t} \equiv S_{i,d}(t)$ and L is a function of d and m.

We now solve the following optimization:

$$\hat{A}, \hat{B} = \arg\min_{A,B} \|(X + AB^T C) - Y\|_F^2,$$

where $A \in \mathbb{R}^{N \times K}$, $B \in \mathbb{R}^{L \times K}$, and $K \leq \min(N, L)$, and C is the spline transformation matrix determined by the given polynomial degree and knots. This problem can be reformulated and solved by reduced rank regression, using the algorithm described in [37]. In this approach, we do not need to apply a penalty to the components of B; the spline basis transformation will ensure that the temporal correction matrix $B^T C$ is smooth.

In this approach, the hyperparameter K is understood the same way as above. The temporal smoothing hyperparameters are different, however. The degree of smoothing is determined by the polynomial degree d and knots t. For simplicity, we will set d as a constant 3; this results in the commonly used cubic spline transformation. We will also enforce that the knots are uniformly spaced, leaving us with the knot interval as the only temporal hyperparameter. The larger the knot interval, the smoother the temporal corrections will be. Note that due to the transformation matrix C, we are no longer able to fit $\tilde{X} = Y$ exactly, even with unbounded K.

We note that we can parameterize the Basis Spline Approach to be equivalent to the Fused Lasso Approach. By setting the basis spline degree to be $d = 0$, the spline transformation matrix C results in a vector that is piecewise constant. If we place a knot at every time point and apply an ℓ_1 penalty to the first differences of the spline components, then the Basis Spline Approach is equivalent to the Fused Lasso Approach. Analogous equivalences hold for higher order splines. If the basis spline degree is $d = 1$, the method is equivalent to trend filtering [31], and so on for higher polynomial degrees.

4.2.4 Preprocessing Indicator Values

All of the models above assume that the heterogeneity corrections should be additive, that is, $\tilde{X} = X + AB^T$. Depending on the application, it may be more reasonable to apply a multiplicative correction. In such a case, we can fit the models using $\log X$ and $\log Y$. If X or Y contain zeros, then we can add a pseudocount and fit using $\log(X + \epsilon)$. We optimize

$$\min_{A,B} \|(\log X + AB^T) - \log Y\|_2^F$$

for the Bounded Rank Model, and the temporal penalties are straightforward for the Fused Lasso and Basis Spline models. Our corrected indicator is $\tilde{X} = X \odot \exp(AB^T)$, where $\odot$ represents the Hadamard product and exponentiation is element-wise. One caveat to note is that the optimization minimizes the mean squared error between the indicator and guide on the log scale.

4.2.5 Hyperparameter Selection

Each of our three models has one or two hyperparameters that control how the guide signal is used. A user may have domain knowledge which suggests that a certain rank is appropriate, in which case, K can be selected manually. A rank could also be selected via various heuristics, such as an elbow plot of the principal components of $Y - X$. Alternatively, multiple values of K could be selected for sensitivity analysis. In this section, we provide a quantitative method of selecting hyperparameters as a default option, as an alternative to manual selection.

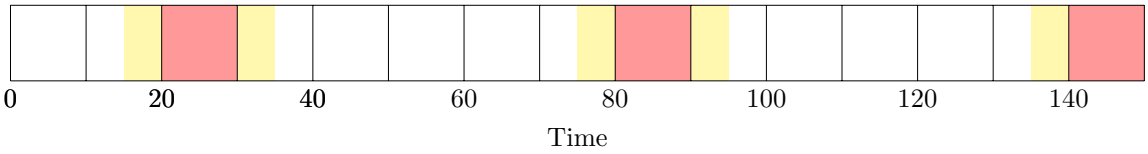

Figure 4.1. We use cross-validation for hyperparameter selection. The red blocks (ten days each) are held out for testing, and the yellow blocks (five days each) are held out to reduce dependencies between the training data and the test data. We repeat for 6 folds.

In our setting, several factors complicate the usually straightforward application of cross validation. First, the data is structured in a two dimensional matrix. Our optimization method does not allow missingness in the matrices, so we cannot simply remove a random subset of data and run the optimization procedure. We can remove entire columns (time points) either randomly or in blocks, but we will need to interpolate the values for the missing time points. We can use mean squared error between $\tilde{X}$ and Y as the error metric, but it is not clear that this is an ideal choice. The indicator and guide measure different quantities and we do not believe or wish that success is defined as matching $\tilde{X}$ and Y.

Despite these challenges, we will select hyperparameters by using a cross validation framework with mean squared error as the error metric. In order to reduce the temporal dependencies inherent in the data, we leave out blocks of time for testing, as illustrated in Fig 4.1. We use linear interpolation to populate the rows of B in the test set, as illustrated in Fig 4.2. Our error metric is the mean squared error between $\tilde{X}$ and Y on the test set.

In the penalized regression context, it is common to apply the "one standard error rule" to cross validation, in which we select the most parsimonious model whose cross validation error is within one standard error of the the minimum cross validation error across all models [27]. A common justification for this rule is that the cross validation errors are calculated with variance, and it is preferable to take a conservative approach against more complex models [27]. Our setup provides further motivation to apply this rule. Unlike in standard cross validation, our goal is not to find the model which fits best to Y, but rather to use Y as a guiding signal to mitigate heterogeneity. Additionally, there is likely a slight dependence between the training data and test data due to the temporal structure of the data. Applying the "one standard error rule" will prevent overadjustment to Y.

In order to use the "one standard error rule", we will need to calculate the number of parameters for a given model. For the Bounded Rank and Basis Spline models, this is straightforward. For the Bounded Rank Model, the number of degrees of freedom is $K(N+T-1)$, and for the Basis Spline Model, it is $K(N+L-1)$, where L is the dimensionality of the basis spline transformation matrix C. For the Fused Lasso Model, we cannot simply calculate the number of entries in the matrices A and B. We will use a result that applies to generalized lasso problems under weak assumptions [49]. In our case, we will estimate the degrees of freedom in matrix B as $\|\Delta_t B\|_0$, or the count of non-zero successive differences along the time axis of B. The total degrees of freedom for the Fused Lasso Model is $K(N-1) + \|\Delta_t B\|_0$.

We note that the theorem in [49] applies only to generalized lasso problems, and in our case, we use an iterative approach of which the fused lasso is just a subroutine. Therefore, the results may not hold precisely in our case. However, we are using the "one standard error rule" merely as a heuristic, and we do not require absolute accuracy in estimating the degrees of freedom.

We reiterate that this is a general rule, and the user can use any rule to select hyperpa-

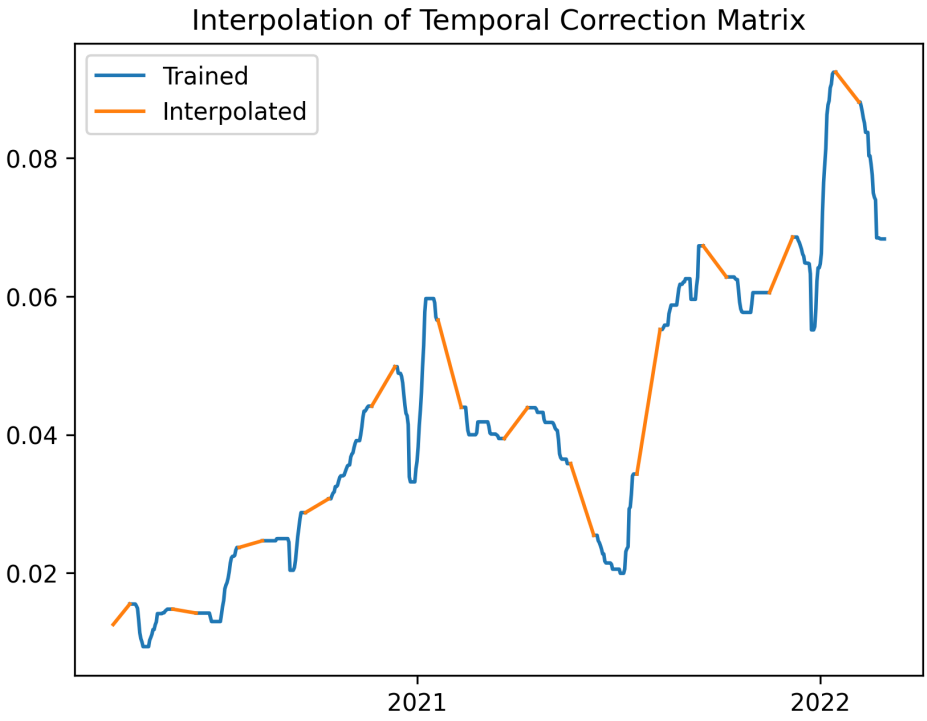

Figure 4.2. We need to interpolate the test indices of the temporal adjustment matrix B in order to calculate $\tilde{X}$. We do this by linear interpolation between the values of B on the boundaries of the blocks of training indices. This figure shows interpolation for a single column of matrix B, as an example.

rameters. If a user has domain knowledge which suggests that a certain rank is appropriate, then they could simply select that rank. If a user wants a more parsimonious model, they could use a two standard error rule or a three standard error rule. In some cases, the cross validation error may have a clear elbow, which could suggest an ideal rank. The "one standard error rule" is used simply as a baseline when no obvious choice exists.

4.3 Results

4.3.1 Simulation Experiments

We first performed experiments on simulated data, where the true rank of the difference matrix $Y - X$ was known. We fit each of the three models to the difference matrix and evaluate performance through cross validation. The simulation setup is as follows:

1. Generate A as a $N \times K$ matrix, where $A_{ij} \overset{\text{iid}}{\sim} \text{Unif}(-1, 1)$.
2. Generate B as a $T \times K$ matrix. For each column k, select nine random breakpoints $(b_1^k, ..., b_9^k)$ between 1 and $T - 1$. Set $b_0^k = 0$ and $b_1^k 0 = T$. Set $B_{b_i^k : b_{i+1}^k, k}$ to be a random constant between 0 and 1. Thus each column of B is piecewise constant with 10 pieces.
3. Let $C = AB^T$, then normalize to have standard deviation 1 across all elements.
4. Let the simulated difference matrix $Y - X$ be D, where $D_{ij} = C_{ij} + \epsilon_{ij}$ and $\epsilon_{ij} \overset{\text{iid}}{\sim} N(0, \sigma = 0.1)$. Note that we do not have to simulate X or Y individually, only the difference.

In our simulations, we used $N = 51$ and $T = 699$ as in our real-world analysis. We experimented with $K \in \{5, 10, 40\}$. For the simulations, we will discuss the Fused Lasso Model (including $\lambda = 0$, which is equivalent to the Bounded Rank Model). The Basis Spline Model does not perform well in the simulations, as will be discussed in the next section.

For $K = 5$, the rank selected by cross validation is 4, as shown in Fig 4.3. In applications where the signal-to-noise ratio is low, our methods will have difficulty in detecting all of the heterogeneity. In this simulation, we can decompose the signal and noise exactly, and perform SVD on the signal and noise separately to determine the signal-to-noise ratio. The first 4 singular values of the signal matrix C are larger than those of the noise matrix ϵ, but the remainder are smaller. We do not see a strong signal in all 5 singular values partially because the rows of AB^T were not constructed to be orthogonal. This supports the hypothesis that the optimal rank was not found due to the low signal-to-noise ratio. We see a similar pattern for $K = 10$ (Fig 4.4), where the rank selected by cross validation is 8 and the first 8 singular values of C are larger than those of ϵ. Once K exceeds the optimal rank, the cross validation error slowly increases, just as would be expected if the model were overfitting.

Fig 4.5 shows that for $K = 40$, the rank with the minimum cross validation error is indeed 40, using the Fused Lasso Model with $\lambda = 1$. Even though the signal-to-noise ratio is higher than 1 for only the first 11 singular values, the correct rank is selected. This is a somewhat surprising result, and might be attributable to the penalty encouraging the rows of B to be piecewise constant. This may allow the model to detect even parts of the signal that are weaker than the noise.

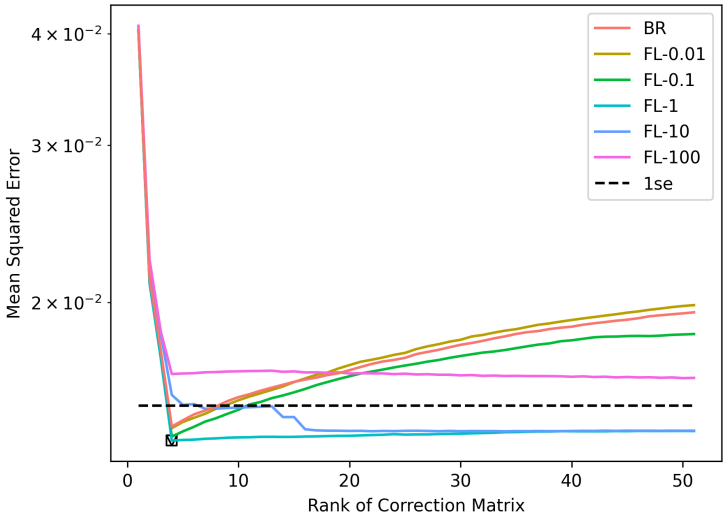

Figure 4.3. Although the true rank of the correction matrix is 5, the optimal rank selected is 4 and $\lambda = 1$. For the Bounded Rank Model (BR) and small values of λ for the Fused Lasso Model (FL), a clear overfitting curve appears. Even when applying the one standard error rule (1se), the same model is selected (as denoted by the square and triangle).

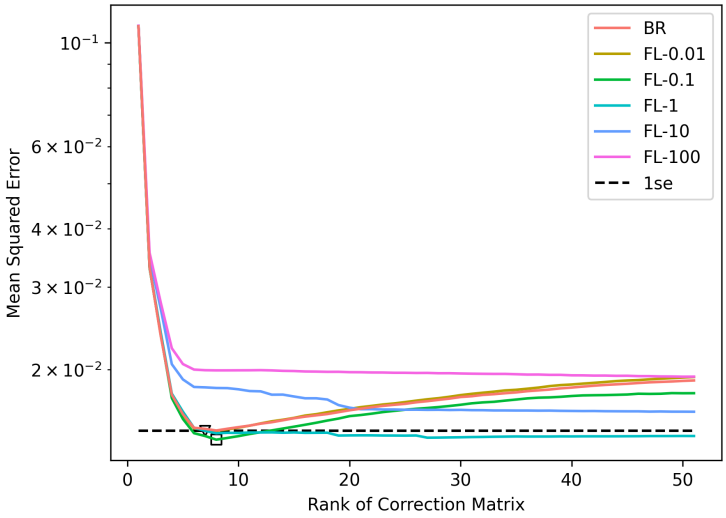

Figure 4.4. Although the true rank of the correction matrix is 10, the optimal rank selected is 8 and $\lambda = 0.1$ (denoted by the square). For the Bounded Rank Model (BR) and small values of λ for the Fused Lasso Model (FL), a clear overfitting curve appears. When applying the one standard error rule (1se), the model selected has rank 7 and $\lambda = 1$ (denoted by the triangle).

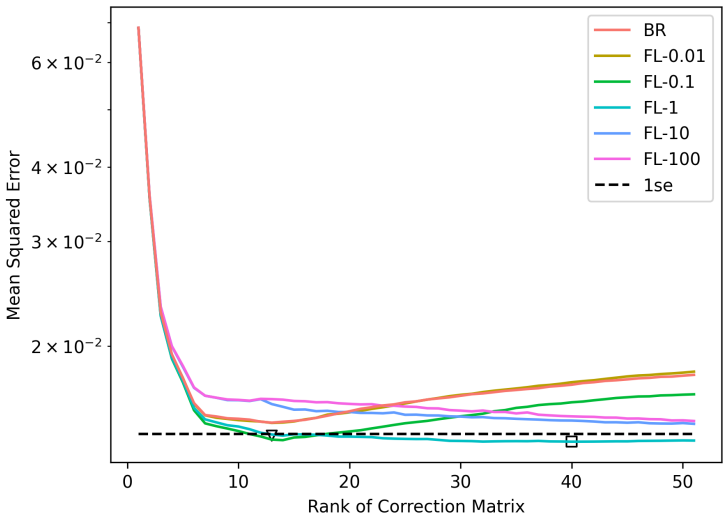

Figure 4.5. The true rank of the correction matrix is 40, and the optimal rank selected is 40 with $\lambda = 1$ (denoted by the square). For the Bounded Rank Model (BR) and small values of λ for the Fused Lasso Model (FL), a clear overfitting curve appears with the minimum significantly lower than the optimal rank. When applying the one standard error rule (1se), the rank selected is 13 with $\lambda = 1$ (denoted by the triangle).

4.3.2 COVID-19 Insurance Claims and Reported Cases

Insurance claims are a useful data source in modeling and forecasting epidemics. They provide information about how many people are sick enough to seek medical care, which is potentially more useful than simply the number of people who are infected but potentially asymptomatic. They can also be available at high geographic and temporal resolution, as well as cover a large proportion of the total population. In this section, we will use a dataset of aggregated insurance claims provided by Optum. The signal is the fraction of all outpatient claims with a confirmed COVID-19 diagnosis code, followed by smoothing and removal of day-of-week effects [15]. Despite the advantages of claims datasets, they are often subject to spatial and temporal heterogeneity, as we will demonstrate.

We used reported COVID-19 cases from Johns Hopkins [16] as our guide signal to correct for heterogeneity in the insurance claims signal. As in the simulation experiments, we used the hyperparameter selection scheme described in Section 4.2.5. Because we believe that the effects of heterogeneity here are multiplicative rather than additive, we applied preprocessing steps as described in Section 4.2.4. We set X to be the log of the insurance claims signal, and Y to be the log of the reported cases signal, each with a pseudocount of $\epsilon = 1$ to account for zeros.

Unlike in the simulation experiments, we do not see a clear overfitting curve. As shown in Fig 4.6, the cross validation error decreases as K increases and λ decreases (as the model's complexity increases) and then flattens. The model with the best cross validation error has $K = 50$, where the rank of the difference matrix is 51. Clearly, we do not want to use this model, since we do not believe that the heterogeneity present in the claims signal has rank 50 out of a possible 51. This is where the "one standard error rule" is useful. It selects the Fused Lasso Model with rank $K = 12$ and $\lambda = 1$. Although this model still has a higher rank

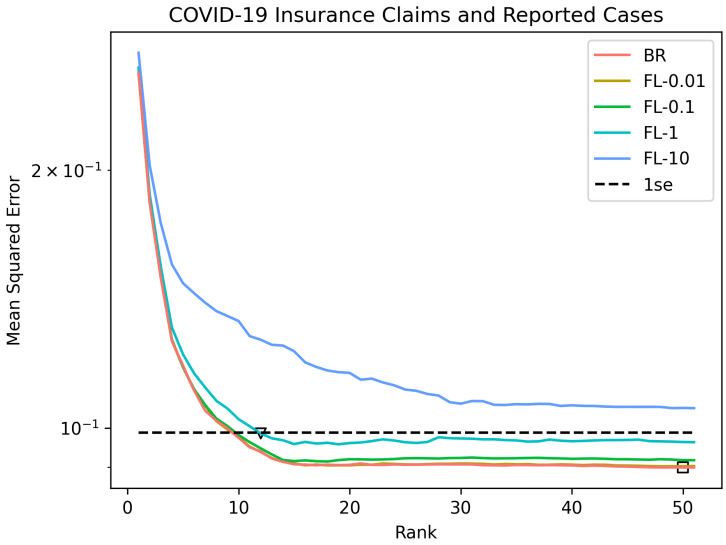

Figure 4.6. Cross validation error is optimized at $K = 50$ using the Bounded Rank Model (BR), i.e. $\lambda = 0$, indicated by the square. However, when applying the one standard error rule, we select the Fused Lasso Model (FL) with $K = 12$ and $\lambda = 1$, indicated by the triangle. This results in a great reduction in parameters with a small decrease in cross validation accuracy.

than we may have thought appropriate, it is much simpler than the model which minimizes cross validation error.

The Basis Spline Models perform poorly on this dataset, as shown in Fig 4.7. For very small knot intervals, some models are candidates for selection under the "one standard error rule", but their degrees of freedom are larger than those of the Fused Lasso Models. We examine the behavior of the Basis Spline Models in Fig 4.8, where we see that there is overfitting if the knot interval is too short (many parameters) and underfitting if the knot interval is too long (fewer parameters). After performing linear interpolation, the overfitting model ends up with reasonable accuracy. However, the basis splines themselves do not accurately represent the temporal corrections. We conclude that the assumption of cubic splines is too rigid in this case. The splines simply cannot fit well to the data, likely due to abrupt changepoints that the Fused Lasso Models are able to handle better.

In Figure 4.9, we illustrate the benefit of applying heterogeneity corrections. The raw insurance claims signal is quite different than the reported case signal in late summer in 2020. The state with the highest claims signal is New York, even though New York has one of the lowest rates of confirmed cases. After applying heterogeneity correction using cases as a guide, the insurance claims signal looks more similar to the reported case signal, improving the comparability of the insurance claims signal across states.

4.3.3 Evaluating Preprocessing Assumptions

As mentioned above, we applied a log transform to the data, assuming that the heterogeneity effects are multiplicative rather than additive. We can test that assumption by comparing the following three models.

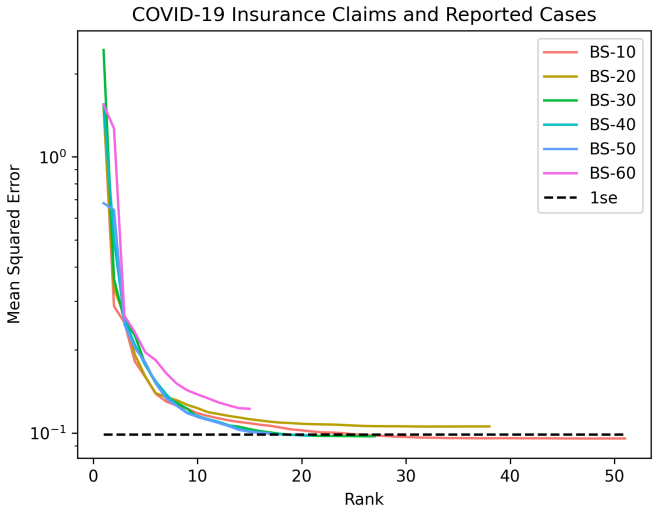

Figure 4.7. We plot the performance of the Basis Spline Model (BS) for different knot intervals. These models have a lower accuracy than the other models, but for some hyperparameters, the Basis Spline Model comes within one standard error of the best Bounded Rank or Fused Lasso Model. Note that the higher the knot interval, the lower the rank of the spline transformation matrix C. For knot intervals more than 10 days, the maximum rank of the model is less than $\min(N, T) = 51$.

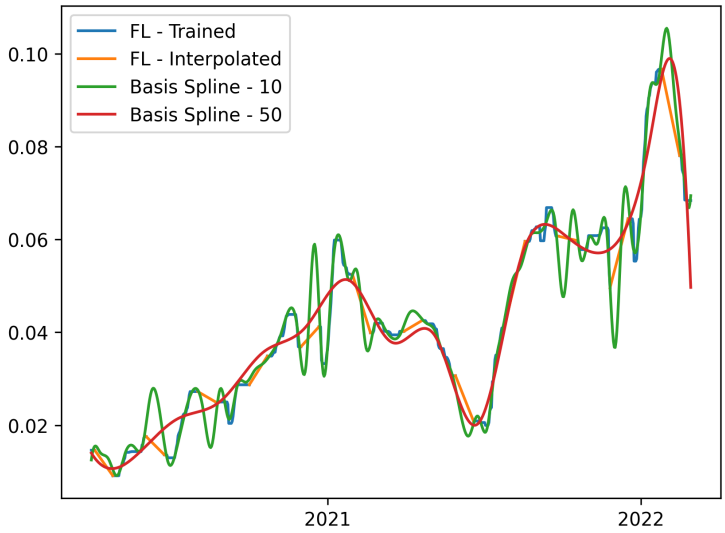

Figure 4.8. We plot the first component of the temporal correction matrix for the Fused Lasso (FL) and Basis Spline models. The orange segments correspond to held-out data that needs to be interpolated for the Fused Lasso Model. The Basis Spline Model with a knot interval of 10 tracks very closely to the Fused Lasso Model in training data but diverges wildly in held-out data. The Basis Spline Model with a knot interval of 50 is smooth but does not have the flexibility to closely match the Fused Lasso Model.

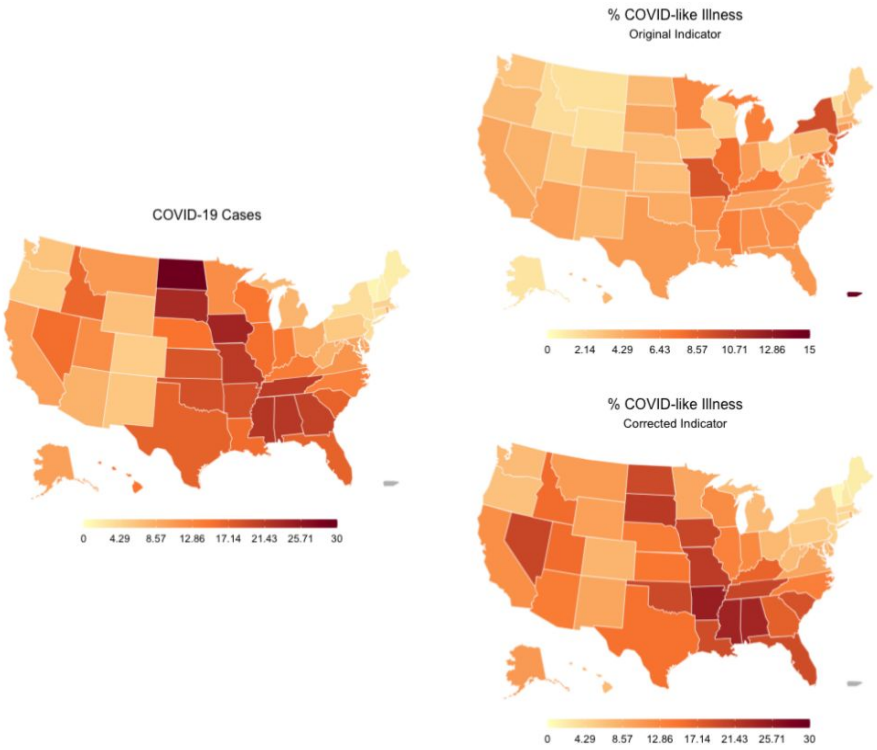

Figure 4.9. Applying a rank-2 correction improves similarity between reported COVID-19 cases and the insurance claims signal. On the left, the average daily confirmed COVID-19 cases between August 15 and September 15, 2020 are displayed in a choropleth map. On the right, we display the value of the insurance claims signal for the same time period before (top) and after (bottom) applying a rank-2 heterogeneity correction using the Bounded Rank Model. The pre-correction %CLI map is not similar to the cases map, but the post-correction %CLI map is.

1. Bounded Rank Model with rank $k = 1$ (BR-1):

$$\min_{a,b} \sum_{i=1}^{N} \sum_{t=1}^{T} (\log X_{it} + a_i \cdot b_t - \log Y_{it})^2$$

2. Additive Model in log space (AL):

$$\min_{a,b} \sum_{i=1}^{N} \sum_{t=1}^{T} (\log X_{it} + a_i + b_t - \log Y_{it})^2$$

3. Additive Model in count space (AC):

$$\min_{a,b} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{X_{it} + a_i + b_t}{Y_{it}} - 1 \right)^2$$

All of these models have $N + T$ parameters and a total of $N + T - 1$ degrees of freedom, with a single parameter for each location and a single parameter for each day, with no regularization. In the first two models, the heterogeneity is assumed to be additive in the log space, or multiplicative in the count space. In the AC model, the heterogeneity is assumed to be additive in the count space. In the BR-1 model, the heterogeneity parameters are multiplied together, whereas in the AL model, they are added. Note that we minimize the relative error for the AC model so that all three models have the same objective.

We display the mean squared error between $\log \tilde{X}$ and $\log Y$ for each of the three models below, with the standard error in parentheses. The models BR-1 and AL, which assume that heterogeneity is multiplicative, perform much better than AC, which assumes that heterogeneity is additive. This supports our initial assumption that the effects of heterogeneity in this particular signal pair are multiplicative.

The AL model performs slightly better than the BR-1 model, which weakly suggests that the spatial and temporal parameters should be added instead of multiplied together. However, the AL model cannot be generalized to higher rank corrections. Therefore, we cannot use this model in practice, as we believe that the effects of heterogeneity are too complex to be modeled solely by a single parameter for each location and for each time point.

Model	BR-1	AL	AC
MSE	0.2205 (0.00220)	**0.2138 (0.00235)**	0.7611 (0.00919)

4.3.4 Google Trends and CTIS Survey

Google has made public an aggregated and anonymized dataset of Google search queries related to COVID-19 [36]. An indicator derived from this dataset roughly measures the relative prevalence of a specific set of search queries in a given location and time. Ideally, this indicator could inform us approximately how many people are symptomatic at a given time at a very minimal cost. However, search query behavior is affected by many other factors other than whether a person is symptomatic. People may be more likely to search for COVID-19 related terms if someone famous was reported as infected, or if public health

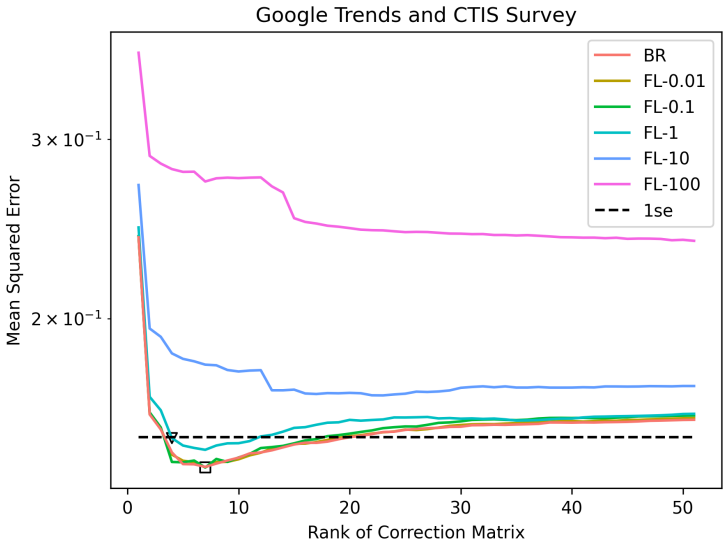

Figure 4.10. When correcting the Google Trends signal, cross validation error is optimized at $K = 7$ using the Fused Lasso Model (FL) with $\lambda = 0.1$, indicated by the square. However, when applying the one standard error rule, we select $K = 4$ and $\lambda = 1$, indicated by the triangle.

measures were enacted or lifted. These can create both spatial and temporal heterogeneity in the indicator.

We used the COVID-19 Trends and Impact Survey (CTIS) as a guide signal. Specifically, our guide is the estimated percentage of people with COVID-like illness from the survey. We used the hyperparameter selection scheme as above.

The results here, shown in Figure 4.10, look more similar to the results in the simulated dataset. The model that performs best in cross validation has a rank of 7, and when applying the one standard error rule, the optimal model is a Fused Lasso Model with rank $K = 4$ and $\lambda = 1$. With increasing K, the cross validation errors increase, indicating that some overfitting can occur. Here as well, the Basis Spline Models perform poorly (not pictured), entrenching a pattern seen in the insurance claims experiment as well.

We examine the temporal components of the optimal model in Fig 4.11. As expected, the components are piecewise constant across time. The first (most important) component is mostly negative in the beginning of the pandemic and spikes during the Omicron wave. By using the CTIS survey as a guide, we correct the Google Trends signal downwards in the beginning of the pandemic and upwards during the Omicron wave.

One possible explanation for this heterogeneity is the decline in public attention and anxiety regarding the COVID-19 pandemic. In the beginning and middle of 2020, many asymptomatic people entered COVID-related searches into Google, resulting in a positively biased signal. Throughout most of 2021, minimal corrections are made and the two signals are at their strongest agreement. During the Omicron wave around the beginning of 2022, our method applies a strong positive correction to the Google Trends signal. According to the CTIS signal, COVID-19 cases are highest at this point, but the Google Trends signal does not increase to the same extent, so a further positive correction is needed. One possible explanation is that fewer symptomatic individuals were appearing in the Google signal, potentially because they were more confident that they indeed had a COVID-19

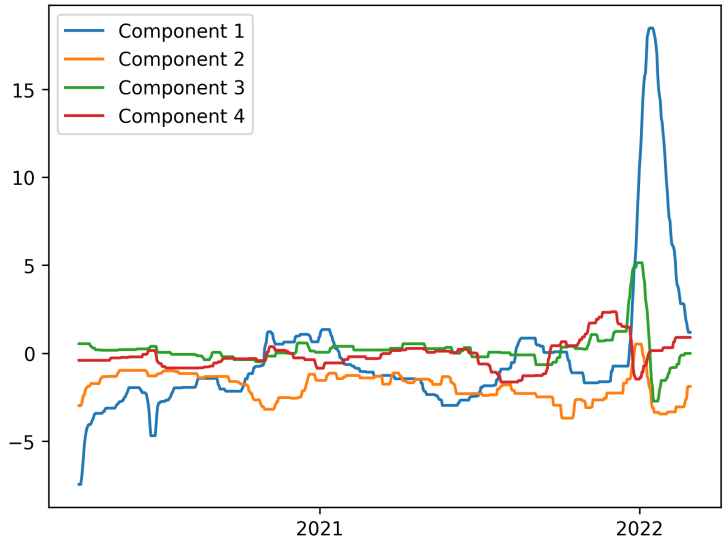

Figure 4.11. We plot the temporal components of the heterogeneity correction between the CTIS and Google Trends signal, using the Fused Lasso model with $K = 4$ and $\lambda = 1$. The most prominent corrections occur around January 2022, corresponding with the Omicron wave.

infection, or because they were less anxious. Another explanation could be that fewer non-symptomatic individuals were appearing in the Google signal, potentially because they were less interested in the pandemic. Whatever the exact reason, the corrections show that the Google signal suffers from temporal heterogeneity, which can be corrected by using the CTIS survey as a guide signal.

4.4 Discussion

As explained above, we define heterogeneity as the presence of location-dependent or time-dependent bias between an indicator and its unobserved ground truth. Indicators are useful sources of information for modeling, mapping, and forecasting epidemics, but conclusions derived from the indicators in the presence of heterogeneity may be suspect. The problem of heterogeneity is poorly suited to translate into an optimization problem in the absence of any ground truth data. Therefore, we use another signal as a guide, and present a method that can use the guide strongly or weakly.

Our method appears to be useful on several pairs of COVID-19 indicators. As Fig 4.9 shows, the raw COVID-19 insurance claims signal gives a very different picture than reported cases. If we were to use the insurance claims signal to understand the current COVID-19 burden across the United States, we could be very misinformed.

The flexibility of our approach is both its main strength and main weakness. On the one hand, the models discussed in this paper can be used for any generic signal and corresponding guide. The user can choose the appropriate parameters based on domain knowledge, exploratory data analysis (e.g. an elbow plot), or the cross validation scheme described above. Because heterogeneity is not straightforward to quantify, we require flexibility to

cover a variety of use cases.

However, this flexibility requires a method to select hyperparameters. In simulations, cross validation yields a reasonable choice of hyperparameters. However, in a real-world setting, the hyperparameters selected by cross validation lead to a model that seems to overadjust. Cross validation might lead to model overadjustment because there are dependencies between the left-out data and the training data. In this case, just as we would expect to overfit in a normal prediction setup, we would expect to overadjust to the guide signal. Additionally, the error metric also encourages overadjustment, since we minimize the squared error between the corrected signal and the guide.

Another significant limitation of this approach is that the guide signal needs to be more reliable than the indicator we are trying to correct. Using Fig 4.9 as an example again, we see that Y is low in New York but X is high, and that after applying our heterogeneity correction, $\tilde{X}$ is low. This is only an improvement if Y is correct, that true COVID-19 activity in New York is actually low. In this case, we have domain knowledge to suggest that reported cases suffer from spatial heterogeneity less than insurance claims. However, were we to treat cases as X and insurance claims as Y, then our "corrected" case signal would be incorrectly high in New York.

An important extension to this approach would be modifying the hyperparameter selection scheme. A better scheme would not default to overadjustment so strongly and would not use an error metric that is optimized when fit exactly to the guide signal. Another extension would be the use of multiple guide signals $Y_1, \ldots, Y_m$. A simple start would be to set $Y = \alpha_1 Y_1 + \cdots + \alpha_m Y_m$ and then apply the heterogeneity correction using Y as the guide signal. Intuitively, if the sources of heterogeneity in the various guides are uncorrelated, then they will tend to cancel out as a result of this averaging, resulting in a spatially and temporally more homogeneous guide. Alternatively, we could view $X, Y_1, Y_2, \ldots, Y_m$ as $m + 1$ different signals, and use them with the models discussed above to jointly estimate the underlying latent quantity to which they are all related. Using multiple guide signals will likely also reduce the overadjustment problem, and a more creative approach to incorporating multiple signals might avoid using the error with the guide signal as a performance metric for hyperparameter selection.

Our current setup fits the adjustment matrix in a batch setting, but a future direction would be to modify the algorithm in an online setting. Indicators are commonly used in real-time, so an online algorithm which makes adjustments as new data arrives may be more appropriate for many use cases.

Of the three models we propose, the Fused Lasso Model performs best in both simulated and real-world experiments. However, it is quite expensive computationally, whereas the other two models can be solved rapidly using SVD-based approaches. Given that the Bounded Rank Model usually performs well, it may be preferable to simply use the Bounded Rank Model in some applications. The Basis Spline Model is slightly more sophisticated without a meaningful increase in computation time. However, the assumptions that lie behind the Basis Spline Model seem to be too strong, specifically when there are abrupt changepoints in temporal heterogeneity.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we provided three examples of epidemiological time series data that suffered from biases or idiosyncrasies. In each case, through a combination of domain knowledge and statistical analysis, we were able to formulate a mathematical definition of the problem, devise a solution, and validate quantitatively that our solution constituted an improvement. In some cases, the lack of ground truth data presented a difficulty in validating our approach. In these cases, we made reasonable assumptions based on domain knowledge that allowed a quantitative comparison between methods.

In each of the examples, our method generalizes well to other types of diseases and data sources. Our recalibration method for forecasts treats the forecaster as a black box, and should work well regardless of the forecasting algorithm used or the forecasting domain or target. The recalibration method detects the types of calibration errors in the original forecasts, assumes that those errors will be present in the current forecasts, and applies a correction. Our method has a hyperparameter that controls for seasonality, which a user can tune depending on the application.

Likewise, the log-linear model to extract an influenza signal from health insurance claims should also generalize well to other diseases. Temporal effects such as day-of-week and holidays are present in a multitude of time series datasets, and in our approach, these effects are learned with minimal assumptions. By assuming that the underlying signal is smooth and that the effects are slowly changing over time, we are able to simultaneously model the temporal effects and fit the underlying signal.

Heterogeneity is a common problem in nearly every dataset, and we present a general solution to correct for it in the absence of ground truth. We specifically make few assumptions about the nature of the heterogeneity in order to present models that can take a wide variety of signals as input. We also provide a general heuristic to selecting the optimal model choice, without relying on domain knowledge.

5.2 Future Directions

The models described in this thesis were all applied to influenza or COVID-19 datasets, whether forecasts, insurance claims, or auxiliary singals. This choice was made due to the

availability of data for these diseases as well as the impact and importance of modeling and forecasting. We hope that these methods will be applied to other diseases in the future, and potentially even other domains. The CDC has identified certain diseases including dengue and norovirus as targets for forecasting, and we believe that these methods will be valuable for those targets.

The challenges presented here extend beyond epidemiology. Miscalibration has been noted in domains such as weather forecasting and economic forecasting, and our recalibration algorithm in Chapter 2 should perform equally well in those domains. Temporal biases such as day-of-week and holiday effects are also common in many datasets, and our method presented in Chapter 3 would likely be successful in extracting a smooth, bias-corrected signal in non-epidemiology datasets as well. Lastly, heterogeneity has been identified in domains such as econometrics, although we are unaware of another approach that attempts to get closer to the ground truth. The approach we present in Chapter 4 may be effective in econometrics as well.

Beyond just applying these methods to other domains, we suggest future directions for improvements to the methods discussed in this thesis. For recalibration, differences in forecaster behavior between the training and testing data is a common problem that inhibits the effectiveness of recalibration. We partially solve this problem by accounting for seasonality, but further work is needed to ensure that the recalibration post-processing actually results in improved forecasts. There is a large gap in the theory regarding the connection between calibration and proper scores, and further work is need to explore this connection. This will ensure that achieving calibration is linked to a guaranteed increase in forecast accuracy.

The methods presented in Chapters 3 and 4 are currently only usable in a retrospective setting, where all of the data has already been observed. While this is useful for analyzing historical data, we are unable to use the corrected signals for forecasting or real-time analysis. Future work would extend these methods to an online setting and correct for the biases in real-time.

We hope that our contribution to heterogeneity correction is the first of many. The problem is common, but it is difficult to validate a solution in the absence of ground truth. Adapting the method to use multiple guide signals is likely to increase the robustness of the solution, and improvements in the hyperparameter selection scheme are likely to do so as well.

Bibliography

[1] Emily L. Aiken et al. "Toward the use of neural networks for influenza prediction at multiple spatial resolutions". In: *Science Advances* 7.25 (2021), eabb1237. DOI: 10.1126/sciadv.abb1237. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abb1237. URL: https://www.science.org/doi/abs/10.1126/sciadv.abb1237.

[2] Heinz H. Bauschke and Jonathan M. Borwein. "Joint and Separate Convexity of the Bregman Distance". In: *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*. Vol. 8. Studies in Computational Mathematics. Elsevier, 2001, pp. 23–36. DOI: https://doi.org/10.1016/S1570-579X(01)80004-5.

[3] Stephen P. Boyd et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122. URL: http://dblp.uni-trier.de/db/journals/ftml/ftml3.html#BoydPCPE11.

[4] Jochen Bröcker. "Reliability, sufficiency, and the decomposition of proper scores". In: *Quarterly Journal of the Royal Meteorological Society* 135.643 (2009), pp. 1512–1519.

[5] Sarah Brocklehurst et al. "Recalibrating software reliability models". In: *IEEE Transactions on Software Engineering* 16.4 (1990), pp. 458–470.

[6] Centers for Disease Control and Prevention. *Epidemic Prediction Initiative: Aedes Forecasting*. https://predict.cdc.gov/post/5e8e21ebcd1fbb050eacaa1e.

[7] Centers for Disease Control and Prevention. *FluSight: Flu forecasting*. https://www.cdc.gov/flu/weekly/flusight/index.html.

[8] Centers for Disease Control and Prevention. *National, regional, and state level outpatient illness and viral surveillance*. https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html.

[9] Vivek Charu et al. "Human mobility and the spatial transmission of influenza in the United States". In: *PLOS Computational Biology* 13.2 (Feb. 2017), pp. 1–23. DOI: 10.1371/journal.pcbi.1005382. URL: https://doi.org/10.1371/journal.pcbi.1005382.

[10] F. Scott Dahlgren et al. "Patterns of seasonal influenza activity in U.S. core-based statistical areas, described using prescriptions of oseltamivir in Medicare claims data". In: *Epidemics* 26 (2019), pp. 23–31. ISSN: 1755-4365. DOI: https://doi.org/10.1016/j.epidem.2018.08.002. URL: https://www.sciencedirect.com/science/article/pii/S1755436518300148.

[11] Benjamin D. Dalziel et al. "Urbanization and humidity shape the intensity of influenza epidemics in U.S. cities". In: *Science* 362.6410 (2018), pp. 75–79. DOI: 10.1126/science.aat6030. eprint: https://www.science.org/doi/pdf/10.1126/science.aat6030. URL: https://www.science.org/doi/abs/10.1126/science.aat6030.

[12] A. P. Dawid. "Calibration-based empirical probability". In: *Annals of Statistics* 13.4 (1985), pp. 1251–1274.

[13] A. P. Dawid. "Statistical theory: The prequential approach (with discussion)". In: *Journal of the Royal Statistical Society Series A* 147 (1984), pp. 278–292.

[14] Carl De Boor. *A practical guide to splines; rev. ed.* Applied mathematical sciences. Berlin: Springer, 2001. URL: https://cds.cern.ch/record/1428148.

[15] *Doctor Visits — Delphi Epidata API*. 2020. URL: https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/doctor-visits.html.

[16] Ensheng Dong, Hongru Du, and Lauren Gardner. "An interactive web-based dashboard to track COVID-19 in real time". In: *The Lancet infectious diseases* 20.5 (2020), pp. 533–534.

[17] Bruno de Finetti. "Methods for Discriminating Levels of Partial Knowledge Concerning a Test Item". In: *The British Journal of Mathematical and Statistical Psychology* 18 (1965), pp. 87–123.

[18] Tillman Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. "Probabilistic forecasts, calibration and sharpness". In: *Journal of the Royal Statistical Society, Series B* 69 (2 2007), pp. 243–268.

[19] Tillman Gneiting and Adrian E. Raftery. "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378.

[20] Tilmann Gneiting and Roopesh Ranjan. "Combining predictive distributions". In: *Electronic Journal of Statistics* 7 (2013), pp. 1747–1782.

[21] Tilmann Gneiting et al. "Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation". In: *Monthly Weather Review* 133 (2005), pp. 1098–1118.

[22] I.J. Good. "Rational decisions". In: *Journal of the Royal Statistical Society* 14 (1 1952), pp. 107–114.

[23] William Greene. "Reconsidering heterogeneity in panel data estimators of the stochastic frontier model". In: *Journal of Econometrics* 126.2 (2005), pp. 269–303.

[24] Thomas M. Hamill and Stephen J. Colucci. "Verification of Eta-RSM short-range ensemble forecasts". In: *Monthly Weather Review* 125 (1997), pp. 1312–1327.

[25] Thomas M. Hamill, Jeffrey S. Whitaker, and Steven L. Mullen. "Reforecasts: An important dataset for improving weather predictions". In: *Bulletin of the American Meteorological Society* 87 (1 2006), pp. 33–46.

[26] Thomas M. Hamill, Jeffrey S. Whitaker, and Xue Wei. "Ensemble reforecasting: Improving medium-range forecast skill using retrospective forecasts". In: *Monthly Weather Review* 132 (2004), pp. 1434–1447.

[27] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[28] Stephen C. Hora. "Probability judgements for continuous quantities". In: *Management Science* 50.5 (2004), pp. 597–604.

[29] *ILI Nearby*. 2016. URL: https://delphi.cmu.edu/nowcast/about.html#HOW_ACCURATE.

[30] William O. Kermack and A. G. McKendrick. "A contribution to the mathematical theory of epidemics". In: *Proceedings of the Royal Society of London* (Aug. 1927). DOI: 10.1098/rspa.1927.0118.

[31] Seung-Jean Kim et al. "ℓ_1 Trend Filtering". In: *SIAM Review* 51.2 (2009), pp. 339–360. DOI: 10.1137/070690274. URL: https://doi.org/10.1137/070690274.

[32] Stephen M. Kissler et al. "Geographic transmission hubs of the 2009 influenza pandemic in the United States". In: *Epidemics* 26 (2019), pp. 86–94. ISSN: 1755-4365. DOI: https://doi.org/10.1016/j.epidem.2018.10.002. URL: https://www.sciencedirect.com/science/article/pii/S1755436517301196.

[33] Levent Kutlu, Kien C. Tran, and Mike G. Tsionas. "A time-varying true individual effects model with endogenous regressors". In: *Journal of Econometrics* 211.2 (2019), pp. 539–559. DOI: https://doi.org/10.1016/j.jeconom.2019.01.014.

[34] Elizabeth C. Lee et al. "Deploying digital health data to optimize influenza surveillance at national and local scales". In: *PLOS Computational Biology* 14.3 (Mar. 2018), pp. 1–23. DOI: 10.1371/journal.pcbi.1006020. URL: https://doi.org/10.1371/journal.pcbi.1006020.

[35] Elizabeth C. Lee et al. "Spatial aggregation choice in the era of digital and administrative surveillance data". In: *PLOS Digital Health* 1.6 (June 2022), pp. 1–12. DOI: 10.1371/journal.pdig.0000039. URL: https://doi.org/10.1371/journal.pdig.0000039.

[36] Google LLC. *Google COVID-19 Search Trends symptoms dataset*. URL: http://goo.gle/covid19symptomdataset.

[37] Ashin Mukherjee and Ji Zhu. "Reduced rank ridge regression and its kernel extensions". In: *Stat. Anal. Data Min.* 4.6 (2011), pp. 612–622. DOI: 10.1002/sam.10138. URL: https://doi.org/10.1002/sam.10138.

[38] World Health Organization. *The top 10 causes of death*. URL: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[39] Pan American Health Organization. *PAHO/WHO Data - Dengue Cases*. https://www3.paho.org/data/index.php/en/mnu-topics/indicadores-dengue-en/dengue-nacional-en/252-dengue-pais-ano-en.html.

[40] Rick Picard and Dave Osthus. "Forecast Intervals for Infectious Disease Models". In: *medRxiv* (2022). DOI: 10.1101/2022.04.29.22274494. eprint: https://www.medrxiv.org/content/early/2022/05/01/2022.04.29.22274494.full.pdf. URL: https://www.medrxiv.org/content/early/2022/05/01/2022.04.29.22274494.

[41] Adrian E. Raftery et al. "Using Bayesian model averaging to calibrate forecast ensembles". In: *Monthly Weather Review* 133 (2005), pp. 1155–1174.

[42] Roopesh Ranjan and Tilmann Gneiting. "Combining probability forecasts". In: *Journal of the Royal Statistical Society, Series B* 72.1 (2010), pp. 71–91.

[43] Nicholas G. Reich et al. "Accuracy of real-time multi-model ensemble forecasts for seasonal influenza in the U.S." In: *PLOS Computational Biology* 15.11 (2019).

[44] 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. DOI: 10.1073/pnas.2111452118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2111452118.

[45] Grant E. Rosensteel et al. “Characterizing an epidemiological geography of the United States: influenza as a case study”. In: *medRxiv* (2021). DOI: 10.1101/2021.02.24.21252361. eprint: https://www.medrxiv.org/content/early/2021/03/01/2021.02.24.21252361.full.pdf. URL: https://www.medrxiv.org/content/early/2021/03/01/2021.02.24.21252361.

[46] Aaron Rumack, Ryan J. Tibshirani, and Roni Rosenfeld. “Recalibrating probabilistic forecasts of epidemics”. In: *PLOS Computational Biology* 18.12 (Dec. 2022), pp. 1–16. DOI: 10.1371/journal.pcbi.1010771. URL: https://doi.org/10.1371/journal.pcbi.1010771.

[47] Aaron Rumack et al. *Recalibration repository*. https://github.com/rumackaaron/recalibration.

[48] Robert Tibshirani et al. “Sparsity and smoothness via the fused lasso”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1 (2005), pp. 91–108. DOI: 10.1111/j.1467-9868.2005.00490.x. URL: http://dx.doi.org/10.1111/j.1467-9868.2005.00490.x.

[49] Ryan J. Tibshirani and Jonathan Taylor. “Degrees of freedom in lasso problems”. In: *The Annals of Statistics* 40.2 (2012), pp. 1198–1232. DOI: 10.1214/12-AOS1003.

[50] Mike Tsionas. *Panel data econometrics : theory*. 1st ed. London: Academic Press, 2019.

[51] Huug van den Dool et al. “The probability anomaly correlation and calibration of probabilistic forecasts”. In: *Weather and Forecasting* 32 (2017), pp. 199–206.

[52] Cécile Viboud et al. “Demonstrating the Use of High-Volume Electronic Medical Claims Data to Monitor Local and Regional Influenza Activity in the US”. In: *PLOS ONE* 9.7 (July 2014), pp. 1–12. DOI: 10.1371/journal.pone.0102429. URL: https://doi.org/10.1371/journal.pone.0102429.

[53] Hung-Jen Wang and Chia-Wen Ho. “Estimating fixed-effect panel stochastic frontier models by model transformation”. In: *Journal of Econometrics* 157 (2010), pp. 286–296.

[54] Daniel S. Wilks and Thomas M. Hamill. “comparison of ensemble-MOS methods using GFS reforecasts”. In: *Monthly Weather Review* 135 (2007), pp. 2379–2390.

[55] Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis”. In: *Biostatistics* 10.3 (Apr. 2009), pp. 515–534. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxp008. URL: https://doi.org/10.1093/biostatistics/kxp008.