Hierarchical nowcasting of counts with an application to COVID-19 hospitalisations in Germany

UseR: Forecasting and nowcasting

Sam Abbott (@seabbs)[1], Felix Günther[2], Johannes Bracher[3], Adrian Lison[4], Sebastian Funk[1]

June 22, 2022

[1] London School of Hygiene Tropical Medicine,
[2] Department of Mathematical Statistics, Stockholm University
[3] Karlsruhe Institute of Technology / Heidelberg Institute for Theoretical Studies
4 ETH Zurich

Table of contents

Introducing the nowcasting problem

Introduction - General problem statement

- Infectious disease data is created by an underlying infection process.
- Infections are generally unobserved.
- We observe other related measures such as the onset of symptoms, test positivity, hospital admission, and death.
- All of these measures happen with some delay from the original date of infection.
- When we observe any of these measures we truncate this delay distribution.

Introduction - Aims of nowcasting

Core aim

Estimate what will ultimately be reported for proxies of infection that we observe with truncation due to their delay from the date of infection.

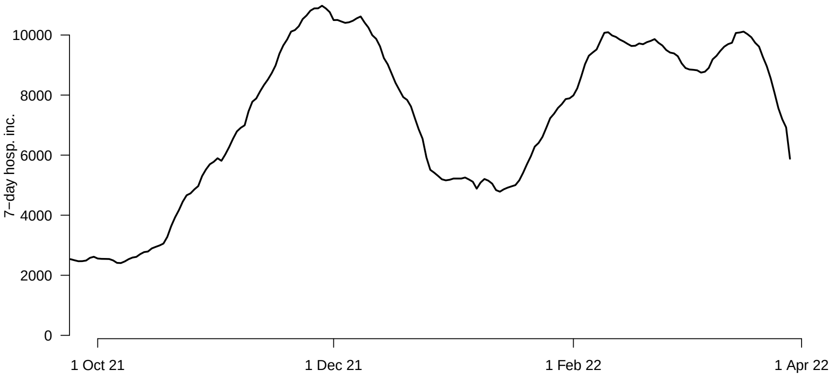
Secondary aims

- Provide improved situational awareness in real-time contexts.
- Estimate the underlying distributions for use in other contexts and to improve understanding of the disease system.
- Improve forecasts of the truncated observations.

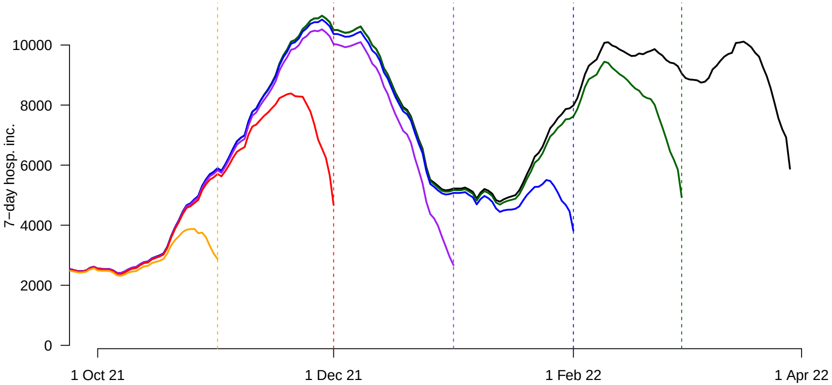
A German example - Seven day hospitalisation incidence

- Definition: The number of persons, who over a seven-day period
  - have been registered electronically as a COVID-19 case by a local health authority (*Meldedatum*).
  - and have been hospitalized (not necessarily during the seven-day period).
- Most recent values are biased downwards due to two types of delays:
  - delay between *Meldedatum* ($\approx$ positive test) and hospitalization.
  - delay between hospitalization and appearance in RKI data.

A German example - What does the data look like?



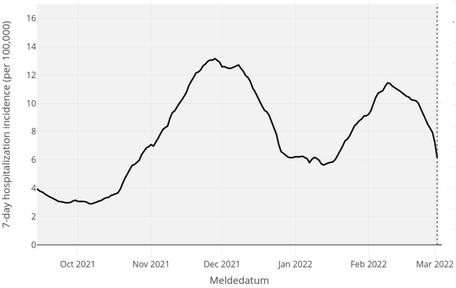
A German example - What does the data look like?



A German example - What are we trying to do?

Goal

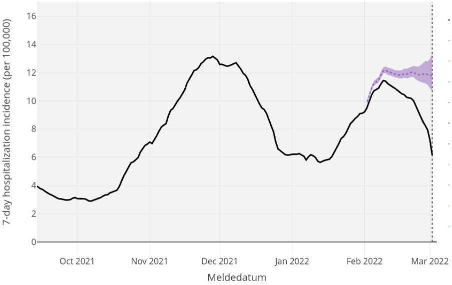
Estimate (predict) what preliminary/incomplete values will ultimately look like.



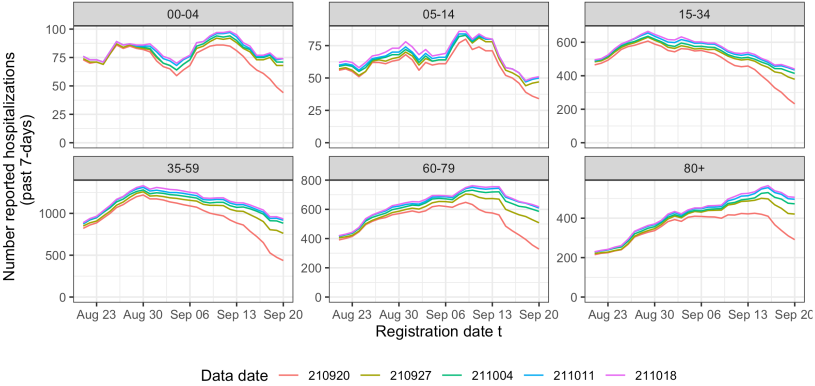
A German example - What are we trying to do?

Goal

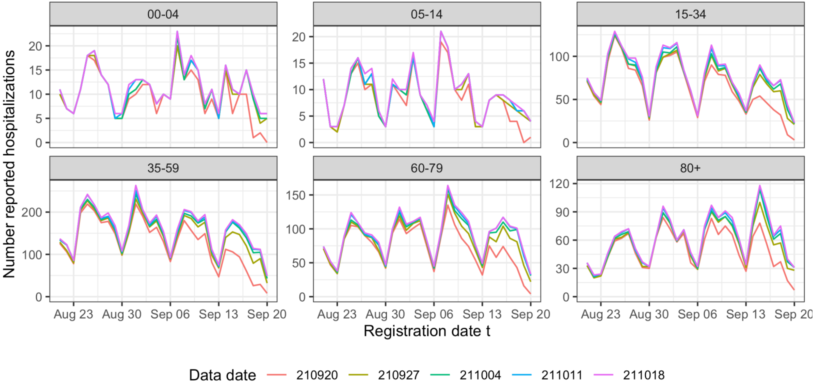
Estimate (predict) what preliminary/incomplete values will ultimately look like.



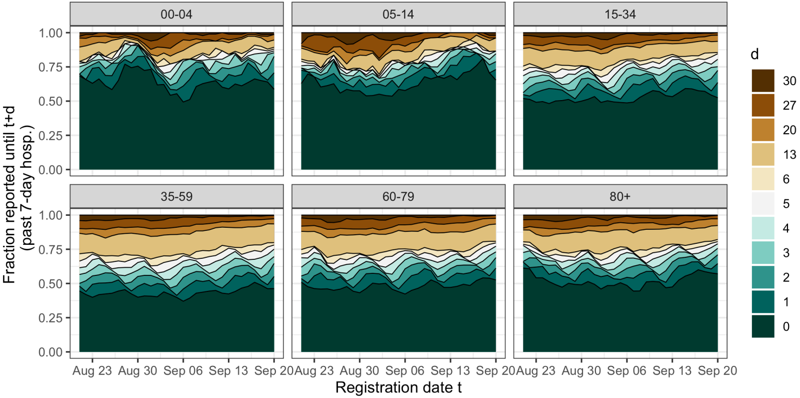
A German example - Age and region stratified



A German example - underlying data has a strong day of week signal



A German example - What about reporting delays?



A German example - Summary

- Seven day hospitalisation incidence by date of positive test is used as a key indicator in Germany.
- These data are truncated and ignoring this may lead to biased surveillance measures and flawed disease making.
- The data is age and location stratified.
- Both incidence and reporting has a strong weekly structure.
- Reporting delays appear to vary over time and by strata.

The nowcasting aim in this context

Estimate hospitalisations for registration days from the number of already reported hospitalisations and the date of these reports.

The Germany nowcasting hub

The Germany nowcasting hub - Multi-model nowcasting

- Experience from weather and infectious disease forecasting (i.e the CDC and ECDC forecasting hubs) shows that combining different models can improve predictions.
- The hub collects and combines probabilistic nowcasts from 8 independently run models.

main hospitalization-nowcast-hub / data-processed / KIT-simple_nowcast / 2022-03-29-KIT-simple_nowcast.csv

dwolffram Update Baseline ✓ Latest comm

1 contributor

5337 lines (5337 sloc) | 472 KB

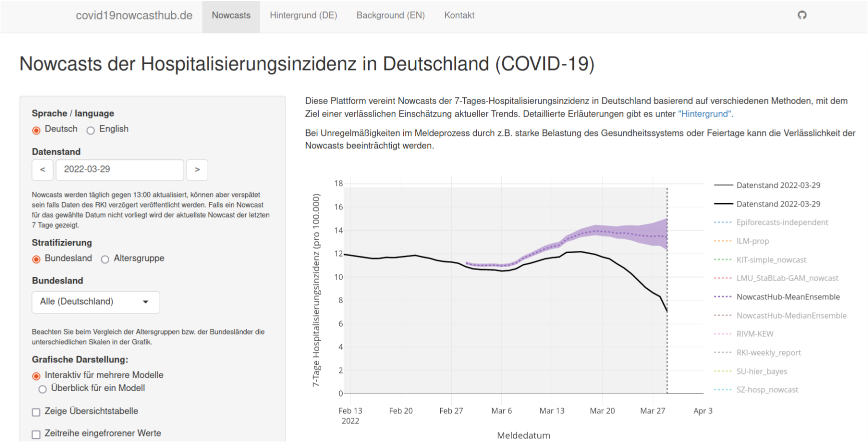
Search this file…

	location	age_group	forecast_date	target_end_date	target	type	quantile	value	pathogen
2	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	mean	NA	10642	COVID-19
3	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.025	8521	COVID-19
4	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.1	9131	COVID-19
5	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.25	9753	COVID-19
6	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.5	10534	COVID-19
7	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.75	11413	COVID-19
8	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.9	12292	COVID-19

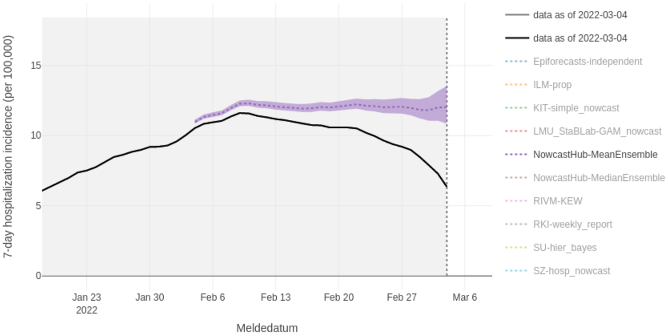
https://github.com/KITmetricslab/hospitalization-nowcast-hub/tree/main/data-truth/COVID-19

The Germany nowcasting hub - Interactive online platform

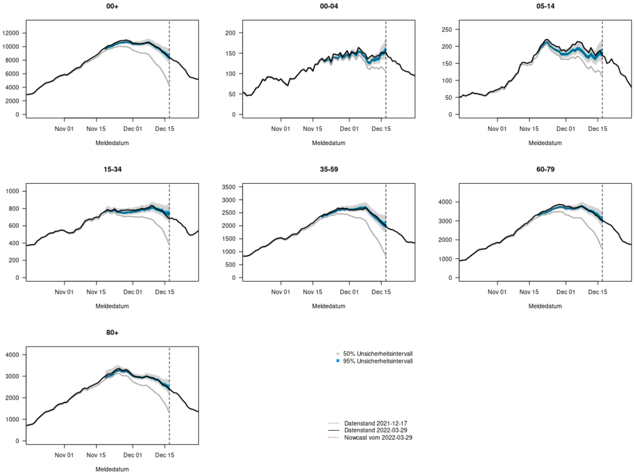
https://covid19nowcasthub.de/



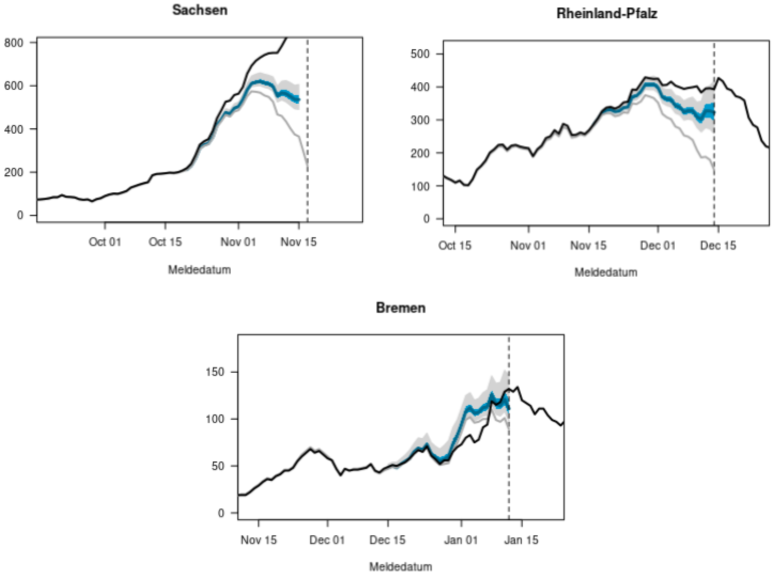
- The main output of the platform is an **ensemble nowcast**, i.e. combination of all available models.



The Germany nowcasting hub - The ensemble is pretty good

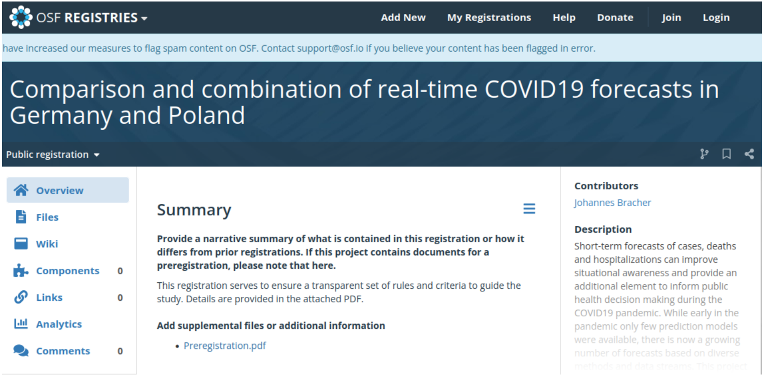


The Germany nowcasting hub - … except when it isn't.



The Germany nowcasting hub - Pre-registered evaluation study

- The hub team are conducting a systematic evaluation study of real-time nowcasts from different methods.
- This study has been pre-registered (*`https://osf.io/mru75/`*) and runs from Nov 2021 through Apr 2022.



A quick tangent into forecast evaluation

Proper scoring rules

The highest expected reward is given if the true probability distribution is supplied as the forecast.

Here the continuous ranked probability score (CRPS) and its approximate cousin the weighted interval score (WIS) are used to evaluate forecasts.

The CRPS is defined as,

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} \left(F(x) - \mathcal{H}(x \geq y)\right)^2 dx$$

F is the CDF, $\mathcal{H}$ is a step function, y is the true value, and x is the forecast.

A quick tangent into forecast evaluation

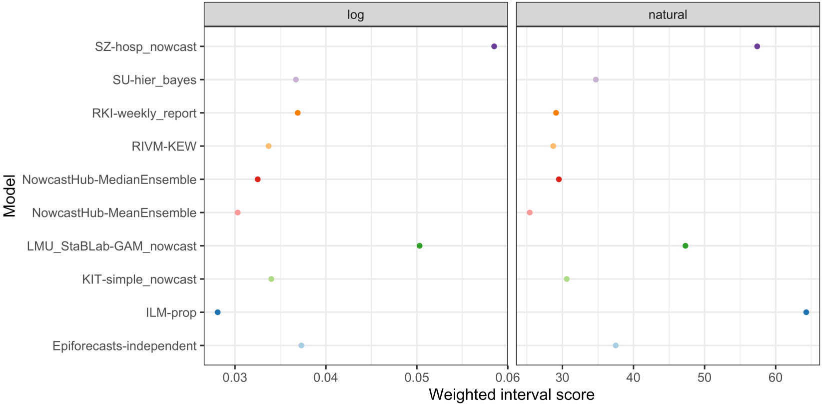
Proper scoring rules

The highest expected reward is given if the true probability distribution is supplied as the forecast.

- A generalisation of absolute error to a probabilistic setting.
- If we take the log of observations and forecasts and calculate the CRPS it becomes an approximate generalisation of the relative error.

The Germany nowcasting hub - Preliminary Evaluation

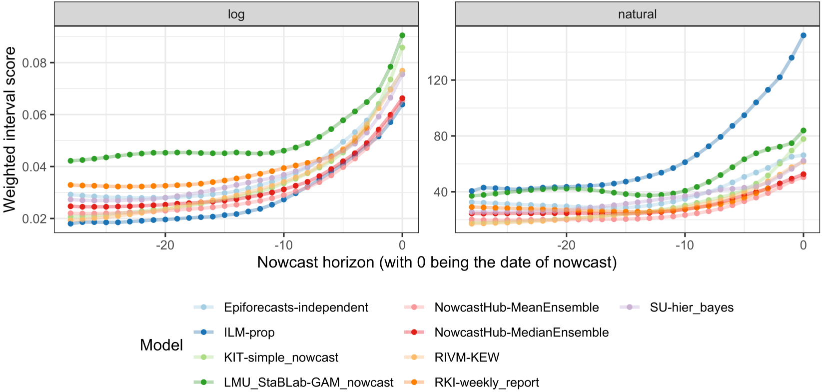
Overall:



See more: *`https://epiforecasts.io/eval-germany-sp-nowcasting/`*

The Germany nowcasting hub - Preliminary Evaluation

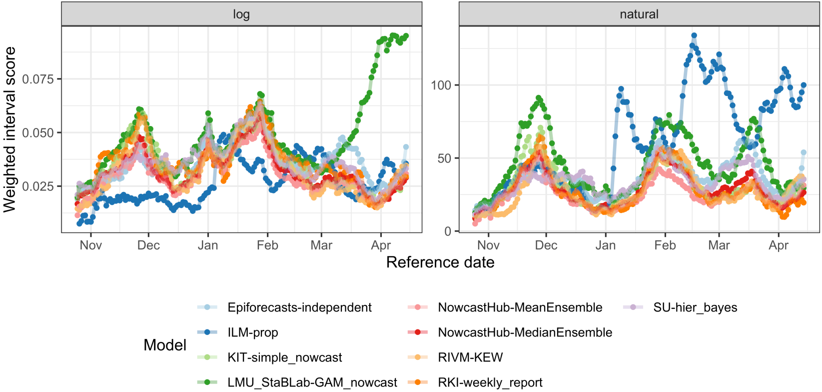
By horizon:



See more: *`https://epiforecasts.io/eval-germany-sp-nowcasting/`*

The Germany nowcasting hub - Preliminary Evaluation

By date of test positivity:



See more : *https://epiforecasts.io/eval-germany-sp-nowcasting/*

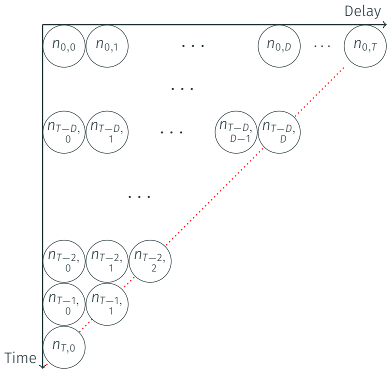
The Germany nowcasting hub - Summary

- In most cases, nowcasts have conveyed a good picture of actual trends.
- Most methods are a bit confident
- Sometimes even the ensemble of all models is very clearly wrong.
- The hub ensemble is generally somewhat better than any individual model
- Collaboratively comparing models allows us to learn about which methods work best.

The epinowcast model

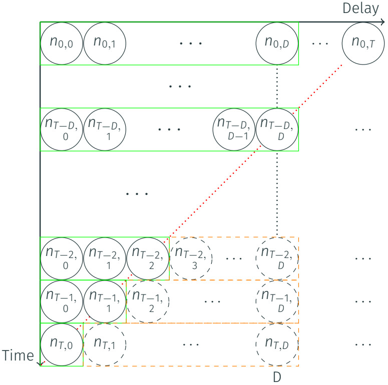
Available data at Day T ('now'), per strata $s = 1 \ldots, S$:

- $n_{t,d,s}$: Count on the reference day t, reported d days after (i.e. at Day $t + d$) for all $t + d \leq T$
- $N_s(t, T) = \sum_{d=0}^{T-t} n_{t,d,s}$: Overall count for reference day t until day T (*now*)



Aim of nowcasting:

- Predict $N_s(t, \infty) = \sum_{d=0}^{\infty} n_{t,d,s}$ for all days $t \leq T$ based on available information at day T
- Corresponds to the prediction of (so far unseen) $N_s(t, \infty) - N_s(t, T)$
- In practice, one defines a maximum reporting delay D, e.g. $D = 35$ days, $N_s(t, \infty) = \sum_{d=0}^{D} n_{t,d,s}$



The epinowcast model - the basic idea

- Building up on Höhle [1], McGough [2], and Günther [3] .
- General idea: separate nowcasting problem into two "sub-models"

  1. Model for the expected number of final notifications
  2. Model for *delay distribution* of the reporting process. This model is further split into 3 sub-models.
     - 2.1 The baseline hazard model.
     - 2.2 The report day hazard model.
     - 2.3 The reference day hazard model.

- When groups of observations (i.e age or location) are present we can either choose to model jointly or independently.

The epinowcast model - expected final notifications sub-model

Here we follow Günther [3] and specify a group specific daily random walk on the log scale. This will soon be generalisable to arbitrary models.

$$
\begin{aligned}
\log(\lambda_{gt}) &\sim \text{Normal}\left(\log(\lambda_{gt-1}), \sigma_g^{\lambda}\right) \\
\log(\lambda_{g0}) &\sim \text{Normal}\left(\log(N_{g0}), 1\right) \\
\sigma_g^{\lambda} &\sim \text{Half-Normal}\left(0, 1\right)
\end{aligned}
$$

Notation

λ_{gt}: expected number of hospitalizations in group g with a reference date at day $t = 0, \ldots, T$

The epinowcast model - delay distribution model

We define the delay distribution (p_{gtd}) as a discrete time hazard model:

$$h_{gtd} = \text{P}\left(\text{delay} = d | \text{delay} \geq d, W_{gtd}\right)$$

We extend this model to decompose W_{gtd} into 3 components:

1. Hazard derived from a parametric delay distribution (γ_{gtd}) dependent on covariates at the date of occurrence.
2. Hazard not derived from a parametric distribution (δ_{gtd}) dependent on covariates at the date of occurrence.
3. Hazard dependent on covariates referenced to the date of report (ϵ_{gtd}).

The epinowcast model - baseline hazard model

We assume that the probability of reporting p'_{gtd} on a given date follows a parametric distribution with the summary parameters defined using reference date indexed fixed (α_i) and random (β_i) coefficients,

$$
\begin{aligned}
p'_{gtd} &\sim \text{LogNormal}\left(\mu_{gt}, \upsilon_{gt}\right) \\
\mu_{gt} &= \mu_0 + \alpha_{\mu} X_{\gamma} + \beta_{\mu} Z_{\gamma} \\
\upsilon_{gt} &= \exp\left(\upsilon_0 + \alpha_{\upsilon} X_{\gamma} + \beta_{\upsilon} Z_{\gamma}\right)
\end{aligned}
$$

The parametric logit hazard for this component of the model is then,

$$
\gamma_{gtd} = \text{logit}\left(\frac{p'_{gtd}}{\left(1 - \sum_{d'=0}^{d-1} p'_{gtd'}\right)}\right)
$$

If we defined this directly using daily hazard terms we would have defined the Cox model.

The epinowcast model - proportional hazard models

We then define our two sub-models that assume proportional hazards.

These act based on the reference date and report date respectively (i.e the first assumes all reports from a given reference day are impacted and the second assumes all reports that occur on a given day are impacted).

Similar these are specified with fixed (α_i) and random (β_i) coefficients.

$$\delta_{gtd} = \mu_0 + \alpha_\delta X_\delta + \beta_\delta Z_\delta \tag{1}$$

$$\epsilon_{gtd} = \epsilon_0 + \alpha_\epsilon X_\epsilon + \beta_\epsilon Z_\epsilon \tag{2}$$

The epinowcast model - Overall hazard and probability of report

The overall hazard for each group, occurrence time, and delay is then,

$$\text{logit}(h_{gtd}) = \gamma_{gtd} + \delta_{gtd} + \upsilon_{gtd},\ h_{gtD} = 1$$

The probability of report for a given delay, occurrence date, and group is then as follows,

$$p_{gt0} = h_{gt0},\ p_{gtd} = \left(1 - \sum_{d'=0}^{d-1} p_{gtd'}\right) \times h_{gtd}$$

The epinowcast model - Observation model

Expected notifications by time of occurrence (t) and reporting delay can now be found by multiplying expected final notifications for each t with the probability of reporting for each day of delay (p_{gtd}).

$$n_{gtd} \mid \lambda_{gt}, p_{gtd} \sim \text{NB}(\lambda_{gt} \times p_{gtd}, \phi),\ t = 1, ..., T.$$

We produce a nowcast of final observed notifications at each occurrence time by summing posterior estimates for each observed notification for that occurrence time.

$$N_{gt} = \sum_{d=0}^{D} n_{gtd}$$

The epinowcast model - Summary

- Phew that was a lot. Can you see why we need a nice and friendly package!
- This is all really a complex regression.
- We can also think of it as a decomposed regression and survival model (i.e Cox and friends).
- The flexible structure outlined here allows us to define a range of models including day of the week effects, random walks by week etc.

The epinowcast R package

The epinowcast R package - Why?

- Previous nowcasting implementations have either been question specific or rigidly defined.
- The nowcasting hub has highlighted the potential complexity of models.
- It has also highlight issues with comparison as there is no easy way for one researcher to run all the models.
- Nowcasting is at the core of many real-time analysis questions but is often not the focus. We want to improve this step for everyone.

The epinowcast R package - What?

An in development R package:

epinowcast 0.0.6.2000 Reference Articles Changelog Search for

Hierarchical nowcasting of right censored epidemological counts

r-universe 0.0.6.2000 License MIT contributors 3

DOI 10.5281/zenodo.5727590

This package contains tools to enable flexible and efficient hierarchical nowcasting of right censored epidemiological counts using a semi-mechanistic Bayesian method with support for both day of reference and day of report effects. Nowcasting in this context is the estimation of the total notifications (for example hospitalisations or deaths) that will be reported for a given date based on those currently reported and the pattern of reporting for previous days. This can be useful when tracking the spread of infectious disease in real-time as otherwise changes in trends can be obfuscated by partial reporting or their detection may be delayed due to the use of simpler methods like truncation

Installation

Installing the package

Install the stable development version of the package with:

```
install.packages("epinowcast", repos = "https://epiforecasts.r-universe.dev")
```

Install the unstable development from GitHub using the following,

```
remotes::install_github("epiforecasts/epinowcast", dependencies = TRUE)
```

Installing CmdStan

If you don't already have CmdStan installed then, in addition to installing `epinowcast`, it is also necessary to install CmdStan using CmdStanR's `install_cmdstan()` function to enable model fitting in `epinowcast`. A suitable C++ toolchain is also required. Instructions are provided in the *Getting started with CmdStanR* vignette. See the CmdStanR documentation for further details and support.

Links

Browse source code

Report a bug

License

Full license

MIT + file LICENSE

Community

Code of conduct

Citation

Citing epinowcast

Developers

Sam Abbott
Author, maintainer

Adrian Lison
Author

More about authors...

Dev status

lifecycle experimental

R-CMD-check passing

codecov 0%

See more: *https://epiforecasts.io/epinowcast/*

The epinowcast R package - What?

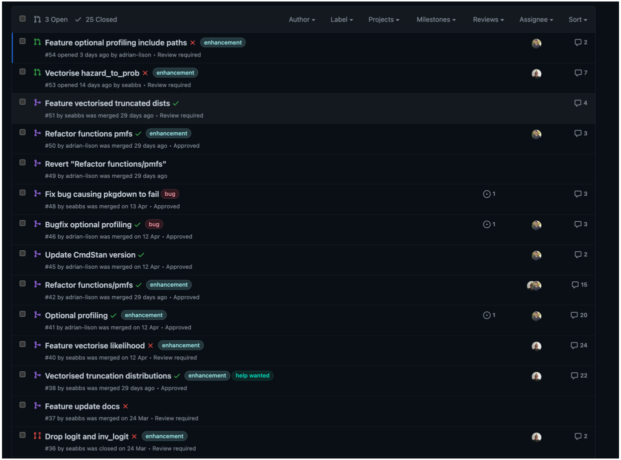
Highly optimised stan implementation:

```
model {
  profile("model_priors") {
  // priors for unobserved expected reported cases
  leobs_init ~ normal(eobs_init, 1);
  eobs_lsd ~ zero_truncated_normal(eobs_lsd_p[1], eobs_lsd_p[2]);
  for (i in 1:g) {
    leobs_resids[i] ~ std_normal();
  }
  // priors for the intercept of the log normal truncation distribution
  logmean_int ~ normal(logmean_int_p[1], logmean_int_p[2]);
  logsd_int ~ normal(logsd_int_p[1], logsd_int_p[2]);
  // priors and scaling for date of reference effects
  if (neffs) {
    logmean_eff ~ std_normal();
    logsd_eff ~ std_normal();
    if (neff_sds) {
      logmean_sd ~ zero_truncated_normal(logmean_sd_p[1], logmean_sd_p[2]);
      logsd_sd ~ zero_truncated_normal(logsd_sd_p[1], logsd_sd_p[2]);
    }
  }
  // priors and scaling for date of report effects
  if (nrd_effs) {
    rd_eff ~ std_normal();
    if (nrd_eff_sds) {
      rd_eff_sd ~ zero_truncated_normal(rd_eff_sd_p[1], rd_eff_sd_p[2]);
    }
  }
  // reporting overdispersion (1/sqrt)
  sqrt_phi ~ normal(sqrt_phi_p[1], sqrt_phi_p[2]) T[0,];
  }
  // log density: observed vs model
  if (likelihood) {
    profile("model_likelihood") {
    target += reduce_sum(obs_lupmf, st, 1, flat_obs, sl, csl, imp_obs, sg, st,
                         rdlurd, srdlh, ref_lh, dpmfs, ref_p, phi);
    }
  }
}
```

See more: *https://github.com/epiforecasts/epinowcast/blob/main/inst/stan/epinowcast.stan*

The epinowcast R package - What?

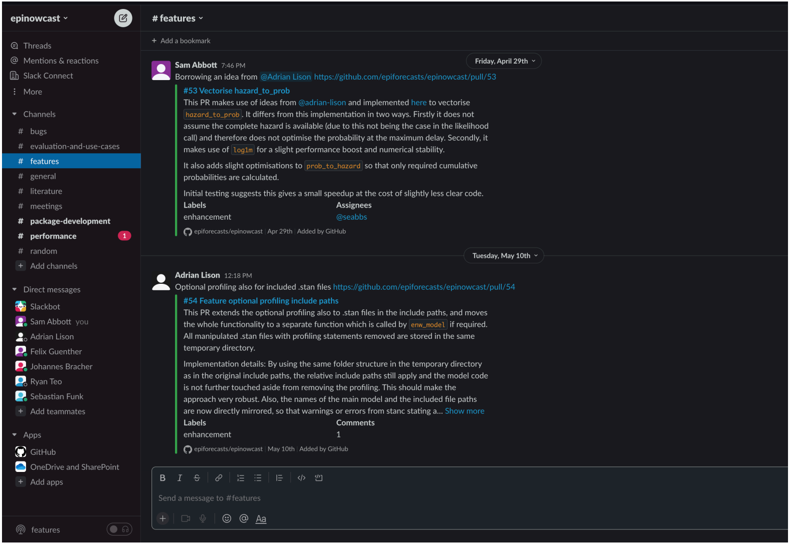
Developed in the open on GitHub:



See more: *https://github.com/epiforecasts/epinowcast/pulls*

The epinowcast R package - What?

An active slack and monthly meeting:



The epinowcast R package - What?

Case studies:

Hierarchical nowcasting of age stratified COVID-19 hospitalisations in Germany

Sam Abbott

Source: vignettes/germany-age-stratified-nowcasting.Rmd

In this vignette we explore using `epinowcast` to estimate COVID-19 hospitalisations by date of positive test in Germany stratified by age using several model specifications with different degrees of flexibility. We then evaluate the resulting nowcasts using visual checks, approximate leave-one-out (LOO) cross-validation using Pareto smoothed importance sampling, and out of sample scoring using the weighted interval score and other scoring measures for the single report date considered here. Before working through this vignette reading the model definition is advised (`vignette("model-definition")`)

Packages

We use the `epinowcast` package, `data.table` and `purrr` for data manipulation, `ggplot2` for plotting, `knitr` to produce tables of output, `loo` to approximately evaluate out of sample performance and `scoringutils` to evaluate out of sample forecast performance.

```
library(epinowcast)
library(data.table)
library(purrr)
library(ggplot2)
library(loo)
library(scoringutils)
library(knitr)
```

This vignette includes several models that take upwards of 10 minutes to fit to data on a moderately equipped laptop. To speed up model fitting if more CPUs are available set the number of threads used per chain to half the number of real cores available (here 2 as we are using 2 MCMC chains and have 4 real cores). Note this may cause conflicts with other processes running on your computer and if this is an issue reduce the number of threads used.

See more: *https://epiforecasts.io/epinowcast/articles/germany-age-stratified-nowcasting.html*

The epinowcast R package - Summary

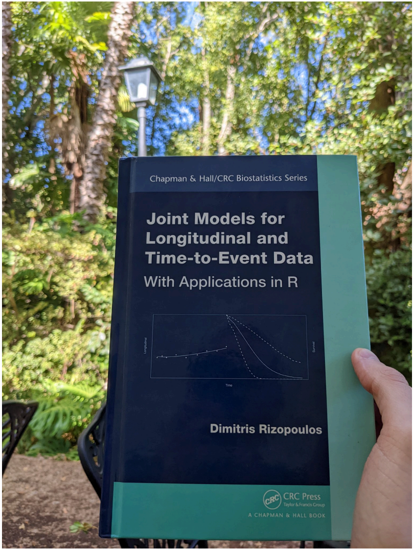
- A flexible nowcasting framework that can fit a range of nowcasting models.
- A community driven package with regular discussions.
- A focus on optimisation and new methodology development.
- Built using software development best practices .
- Evaluated in real-time as part of the Germany nowcasting hub

Extensions

Extensions - Coming soon

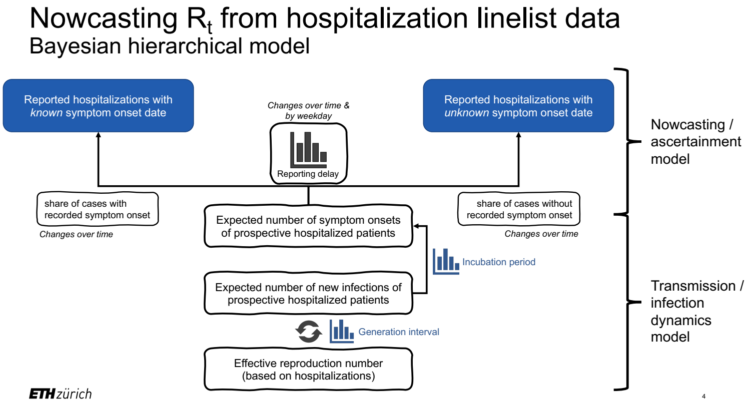
- A full featured formula interface.
- A flexible expectation model.
- The ability to forecast into the future.
- An extension to model missing data from Adrian Lison.
- More software development (help needed).

Extensions - Connection to survival models



Extensions - Latent infection modelling using renewal equations

Creating a one stop shop for situational awareness. Spearheaded by Adrian Lison.



Summary

Summary

- Because of delays between infections and observation truncation is everywhere when studying infectious disease models.
- Multiple models exist to account for this and most perform quite well.
- In general though these models are not used as part of wider practice.
- The epinowcast R package aims to change that.
- There is lots of interesting development to be done and lots of exciting use cases and extensions. Please reach out if interested!

Michael Höhle and Matthias an der Heiden.
Bayesian nowcasting during the STEC O104: H4 outbreak in Germany, 2011.
Biometrics, 70(4):993–1002, 2014.

Sarah F McGough, Michael A Johansson, Marc Lipsitch, and Nicolas A Menzies.
Nowcasting by Bayesian Smoothing: A flexible, generalizable model for real-time epidemic tracking.
PLoS computational biology, 16(4):e1007735, 2020.

Felix Günther, Andreas Bender, Katharina Katz, Helmut Küchenhoff, and Michael Höhle.
Nowcasting the COVID-19 pandemic in Bavaria.
Biometrical Journal, 63(3):490–502, 2021.