Epinowcast: Flexible hierarchical nowcasting of infectious disease surveillance data

Sam Abbott Johannes Bracher Sebastian Funk

Introduction

The `epinowcast` package aims to be a modular toolbox for real-time infectious disease surveillance both in outbreak and routine contexts. As such we provide a modular modelling framework that is optimised for a range of common surveillance tasks whilst maintaining flexibility and supporting user extension.

We provide a flexible semi-parametric model for the underlying generative process similar to that implemented in other real-time infectious disease modelling packages (Abbott, Hellewell, Sherratt, et al. 2020; Scott et al. 2020). This optionally includes a renewal process (Fraser 2007; Cori et al. 2013) and latent reporting process (Abbott, Hellewell, Sherratt, et al. 2020; Abbott, Hellewell, Thompson, et al. 2020; Bhatt et al. 2020). Combined with the appropriate generation time distribution this approach has been shown to correspond to a Susceptible-Exposed-Infected-Recovered (SEIR) model (Champredon, Dushoff, and Earn 2018) with the addition of reporting delays. Howover, our default model contains minimal mechanism in order to more flexibly fit highly informative data.

Our nowcasting approach is an extension of that proposed by Günther et al.(Günther et al. 2021) which was itself an extension of the model proposed by Höhle and Heiden(Höhle and Heiden 2014) and implemented in the `surveillance` R package(Meyer, Held, and Höhle 2017). Compared to the model proposed in Günther et al.(Günther et al. 2021), `epinowcast` adds support for jointly nowcasting multiple related datasets, a flexible formula interface allowing for the specification of a large range of models, and an optional parametric assumption for the underlying reporting delay.

We also support flexible joint modelling of missing data by assuming that the reporting delay is consistent between reported and unreported observations following the methodology of Lison et al. (Lison, n.d.).

Our modelling framework is implemented in the `stan` probalistic programming language via `cmdstanr` (Team 2021; Gabry and Češnovar 2021) with a focus on computational efficiency and robustness.

In the following sections we outline our modelling methodolgy, current feature set stratified by module, highlight implementation details, and finally show an example use case using German hospitalisation data. For each model module we present both our default implementation as well as the more generic framework we support.

Methods

Model

Decomposition into expected final notifications and report delay components

We are concerned with outcomes that occur at a time of *reference* (e.g., date of symptom onset or test for a disease) that are reported only with a delay, at the time of *report* (e.g. the date onsets are entered into a central database and so become available for analysis). We assume that these times are measured in discrete time steps, usually of a day (in which case the times are dates).

We follow the approach of Höhle and Heiden(Höhle and Heiden 2014) and consider the distribution of notifications ($n_{g,t,d}$) by time of reference (t) and reporting delay (d) conditional on the final observed count $N_{g,t}$ for each dataset (g) such that,

$$N_{g,t} = \sum_{d=0}^{D} n_{g,t,d} \tag{1}$$

where D represents the maximum delay between time of reference and time of report which in theory could be infinite but in practice we set to a finite value in order to make the model identifiable and computationally feasible. For each t and g these notifications are assumed to be drawn from a multinomial distribution with $N_{g,t}$ trials and a probability vector ($p_{g,t,d}$) of length D. The aim of nowcasting is to predict the final observed counts $N_{g,t}$ given information available up to time t. We do this by estimating the components of this probability vector jointly with the expected number of final notifications ($\lambda_{g,t} = \mathbb{E}[N_{g,t}]$) in dataset g at time t.

An alternative approach would be to consider each $n_{g,t,d}$ independently at which point the model can be defined as a regression that can be fit using standard software with the appropriate observation model and adjustment for reporting delay (i.e it becomes a Poisson or Negative Binomial regression). An implementation of this approach is available in Bastos et al.(Bastos et al. 2019). A downside of this simplified approach is that reporting is not conditionally dependent which may make specifying models for complex reporting distributions difficult.

Expected final notifications

Default model

Here we follow the approach of Günther et al.(Günther et al. 2021) and specify the model for expected final notifications as a first order random walk. This simple model is highly flexible and so a good fit for nowcasting problems where the data is highly informative.

$$\log(\lambda_{g,t}) \sim \text{Normal}\left(\log(\lambda_{g,t-1}), \sigma_g^{\lambda}\right) \tag{2}$$
$$\log(\lambda_{g,0}) \sim \text{Normal}\left(\log(N_{g,0}+1), 1\right) \tag{3}$$
$$\sigma_g^{\lambda} \sim \text{Half-Normal}\left(0, 1\right) \tag{4}$$

where N_{g0}, the first time point for expected observations in dataset d, is assumed to have been completely observed.

Generalised model

In settings where data is sparse or where users want to understand the underlying generative process our flexible default model is likely not a good choice. In these settings our generic model offers a range of options that are context specific. Our generic model is currently based on a renewal process (Fraser 2007; Cori et al. 2013) with additional latent reporting delays (Abbott, Hellewell, Sherratt, et al. 2020; Abbott, Hellewell, Thompson, et al. 2020; Bhatt et al. 2020). As previously noted (Champredon, Dushoff, and Earn 2018), this corresponds to the commonly used Susceptible-Exposed-Infected-Recovered (SEIR) model when appropriate generation time is specified (Champredon, Dushoff, and Earn 2018)

Instantaneous reproduction number/growth rate

We model the instantaneous reproduction number (R_t) on the log scale (though support for other link functions is planned). When the generation time is fixed to be a day this can be interpreted as the instantaneous growth rate (r_t) defined as the difference in the log of the expected number of final notifications between time t and $t-1$.

$$\log(R_{g,t}) = r_0 + \beta_{f,r} X_r + \beta_{r,r} Z_r \tag{5}$$

where r_0 is the optional intercept, X_r is the design matrix for fixed effects ($\beta_{f,r}$), and Z_r is the design matrix for random effects ($\beta_{r,r}$. Within this specification the default model can be specified as a random effect on the day with no intercept. Alternative specifications that may be of interest include a weekly random walk (specifed as `~ 1 + rw(week)`), a piecewise linear model (specified as `~ 1 + day:week`), and a group specific random effect (specified as `~ 1 + (1 | .group)`).

Latent infections or notifications

We model the expected number of infections or latent notifications (λ^l) using a renewal process (Fraser 2007; Cori et al. 2013). This model is a generalisation of the default model and can be used to model the expected number of latent notifications in a setting where the generation time is not fixed to be a day. It implies that current infections/notifications are dependent on past infections/notifications based on a kernal (usually interpreted as the generation time or serial interval). An instantaneous daily growth rate model can be recovered by setting the generation time to be fixed at 1 day. The model is defined as follows,

$$\lambda^l_{g,t} \sim \text{LogNormal}\left(\mu^l_{g,t}, \sigma^l_{g,t}\right), \ t \leq P \tag{6}$$

$$\lambda^l_{g,t} = R_{g,t} \sum_{p=1}^{P} G_g\left(p, t-p\right) \lambda^l_{g,t-p} \tag{7}$$

Latent reporting delay and ascertainment

In some settings there may be additional reporting delays on top of those that are directly observed in the data, and therefore "nowcastable", a common example is the delay from exposure to symptom onset. For these settings we support modelling "latent" reporting delays as a convolution of the underlying expected counts with the potential for these delays to vary over time and by group. This implementation is similar to that implemented in `EpiNow2` and `epidemia` as well as other similar models (Abbott, Hellewell, Sherratt, et al. 2020; Abbott, Hellewell, Thompson, et al. 2020; Bhatt et al. 2020; Lison, n.d.). In addition to this we support modelling ascertainment through the use of improper probability mass functions (i.e. by not enforcing a sum to 1 constraint) and inferring ascertainment where possible (for example day of the week reporting patterns).

$$\lambda_{g,t} = \nu_{g,t} \sum_{\tau=0}^{L-1} F_g\left(\tau + 1, t-\tau\right) \lambda^l_{g,t-\tau} \tag{8}$$

$$\nu_{g,t} = \nu_0 + \beta_{f,\nu} X_\nu + \beta_{r,\nu} Z_\nu \tag{9}$$

Where $\nu_{g,t}$ is the inferred ascertainment and is modelled flexibly using an optional intercept (ν_0), a design matrix (X_ν) for fixed effects ($\beta_{f,\nu}$), and a design matrix (Z_ν) for random effects ($\beta_{r,\nu}$.

Delay distribution

Where data is available on the report date for a given individual we can estimate the delay distribution directly and jointly rather than relying on estimates from other sources (though we may wish to augment our priors with these estimates). In the following section we describe our default parametric delay distribution model as well as our highly flexible discrete time to event based generic model.

Default model

In our default model we consider the delay distribution to follow a discretised log-normal as follows,

$$p_{g,t,d} \sim \text{LogNormal}\left(\mu^d, \sigma^d\right) \tag{10}$$
$$\mu^d \sim \text{Normal}\left(0, 1\right) \tag{11}$$
$$\sigma^d \sim \text{Half-Normal}\left(0, 1\right) \tag{12}$$

Generalised model

We generalise this model in order to support a range of delay distributions and reporting processes following the approach of Günther et al.(Günther et al. 2021) we estimate the delay distribution ($p_{g,t,d}$) using a discrete-time logistic hazard model

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

but we extend this model to decompose $W_{g,t,d}$ into 3 components: hazard derived from a parametric delay distribution ($\gamma_{g,t,d}$) dependent on covariates at the time of reference, hazard not derived from a parametric distribution ($\delta_{g,t,d}$) dependent on covariates at the time of reference, and hazard dependent on covariates referenced to the time of report ($\epsilon_{g,t,d}$).

The first component ($\gamma_{g,t,d}$) we estimate what would be the probability of reporting $p'_{g,t,d}$ at a given time if it followed a parametric distribution, here implemented using a discretised log normal (a range of other distributions are supported by the package) with the log mean and log standard deviation defined using an intercept and arbitrary shared, reference time indexed, covariates with fixed ($\beta_{f,i}$) and random ($\beta_{r,i}$) coefficients (note these can include auto-regressive terms),

$$p'_{g,t,d} \sim \text{LogNormal}\left(\mu_{g,t}, \upsilon_{g,t}\right) \tag{13}$$
$$\mu_{g,t} = \mu_0 + \beta_{f,\mu} X_\gamma + \beta_{r,\mu} Z_\gamma \tag{14}$$
$$\log(\upsilon_{g,t}) = \upsilon_0 + \beta_{f,\upsilon} X_\gamma + \beta_{r,\upsilon} Z_\gamma \tag{15}$$

Note we normalise this distribution so that it sums to 1. The parametric logit hazard (i.e. the probability of report at a given time conditional on not already having reported) for this component of the model is then,

$$\gamma_{g,t,d} = \text{logit}\left(\frac{p'_{g,t,d}}{\left(1 - \sum_{d'=0}^{d-1} p'_{g,t,d'}\right)}\right) \tag{16}$$

In addition to parametric reporting effects there may also be non-parametric effects referenced by both reference and report dates. These are represented by the non-distributional logit hazard components for the time of reference and report, defined using an intercept (δ_0) and arbitrary shared covariates with fixed ($\beta_{f,i}$) and random ($\beta_{r,i}$) coefficients (note these can include auto-regressive terms).

$$\delta_{g,t,d} = \delta_0 + \beta_{f,\delta} X_\delta + \beta_{r,\delta} Z_\delta \tag{17}$$

$$\epsilon_{g,t,d} = \beta_{f,\epsilon} X_\epsilon + \beta_{r,\epsilon} Z_\epsilon \tag{18}$$

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

$$\text{logit}(h_{g,t,d}) = \gamma_{g,t,d} + \delta_{g,t,d} + \epsilon_{g,t,d},\ h_{g,t,D} = 1 \tag{19}$$

where the hazard on the final day has been assumed to be 1 in order to enforce the constraint that all reported observations are reported within the specified maximum delay. The probability of report for a given delay, reference time, and group is then as follows,

$$p_{g,t,0} = h_{g,t,0},\ p_{g,t,d} = \left(1 - \sum_{d'=0}^{d-1} p_{g,t,d'}\right) \times h_{g,t,d} \tag{20}$$

All ($\beta_{f,i}$) and random ($\beta_{r,i}$) coefficients have standard normal priors by default with standard half-normal priors for pooled standard deviations. For further implementation details see `enw_reference()` for delays linked to the date of reference, `enw_report()` for delays linked to the date of report.

Observation model and nowcast

Expected notifications by time of reference (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 ($p_{g,t,d}$). We assume a negative binomial observation model, by default, with a joint overdispersion parameter (with a standard half normal prior on 1 over square root of the overdispersion(Team 2020)) and produce a nowcast of final observed notifications at each reference time by summing posterior estimates for unobserved notification and observed notifications for that reference time.

$$n_{g,t,d} \mid \lambda_{g,t}, p_{g,t,d} \sim \text{NB}\left((1-\alpha_{g,t})\lambda_{g,t} \times p_{g,t,d}, \phi\right),\ t = 1, ..., T. \tag{21}$$

$$\frac{1}{\sqrt{\phi}} \sim \text{Half-Normal}(0, 1) \tag{22}$$

$$N_{g,t} = \sum_{d=0}^{D} n_{g,t,d} \tag{23}$$

Where $\alpha_{g,t}$ is the proportion of cases by reference date that will not report their reference date. By default this is not modelled and is set to zero , see the accounting for reported cases with a missing reference date section for further defaults. Other observation models such as the Poisson distribution are also supported. See the documentation `enw_obs()` for details.

In order to make best use of observed data when nowcasting we use observations where available and where they have not been reported for a given report and reference date we use the posterior prediction from the observation model above. This means that as nowcast dates become increasingly truncated they depend more on modelled estimates whereas when they are more complete the majority of the final count is known. Depending on your use case the posterior predictions alone may also be of interest.

Accounting for reported cases with a missing reference date

In real-world settings observations may be reported without a linked reference date. A common example of this is cases by date of symptom onset where report date is often known but onset date may not be. To account for this we support modelling this missing process by assuming that cases with a missing reference date have the same reporting delay distribution as cases with a known reference date and that processes that drive the probability of having a missing reference date ($ _{g,t}) are linked to the unknown date of reference rather than the date of report based on Lison et al. (Lison, n.d.). We model this probability flexibly on a logit scale as follows,

$$\text{logit}(\alpha_{g,t}) = \alpha_0 + \beta_{f,\alpha} X_\alpha + \beta_{r,\alpha} Z_\alpha \tag{24}$$

Where α_0 represents the intercept, $\beta_{f,\alpha}$ fixed effects, and $\beta_{r,\alpha}$ random effects. To link with observations by date of report with a missing reference date ($M_{g,t}$) we convolve expected notifications with the probability of having a missing reference date and the probability of reporting on a given day as follows,

$$M_{g,t} \mid \lambda_{g,t}, p_{g,t,d}, \alpha_{g,t} \sim \text{NB}\left(\sum_{d=0}^{D} \alpha_{g,t-d}\lambda_{g,t-d}p_{g,t-d,d}, \phi\right), \ t = 1, ..., T. \tag{25}$$

As for cases with known reference dates other observation models are supported. For further implementation details see `enw_missing()`.

Implementation

The model is implemented in the probalistic programming language `stan` and we use `cmdstanr` to interact with the model (Team 2021; Gabry and Češnovar 2021). Optional within chain parallelisation is available across times of reference to reduce runtimes. Sparse design matrices have been used for all covariates to limit the number of probability mass functions that need to be calculated. `epinowcast` incorporates additional functionality written in R(R Core Team 2019) to enable plotting nowcasts and posterior predictions, summarising nowcasts, and scoring them using `scoringutils`(Bosse 2020). A flexible formula interface is provided to enable easier implementation of complex user specified models without interacting with the underlying code base. All functionality is modular allowing users to extend and alter the underlying model whilst continuing to use the package framework.

Results

Discussion

Summary

Strengths and Weaknesses

Literature context

Further work

Conclusions

Acknowledgments

We thank Molly for being a good Labrador.

Data availability

All data and code are available here:

https://github.com/seabbs/epinowcast-MLGHW-2023

Funding

SA, and SF were funded by a Wellcome senior fellowship to SF (210758/Z/18/Z). JB acknowledges support from the Helmholtz Foundation via the SIM-610 CARD Information and Data Science Pilot Project.

References

Abbott, Sam, Joel Hellewell, Katharine Sherratt, Katelyn Gostic, Joe Hickson, Hamada S. Badr, Michael DeWitt, Robin Thompson, EpiForecasts, and Sebastian Funk. 2020. *EpiNow2: Estimate Real-Time Case Counts and Time-Varying Epidemiological Parameters*. https://doi.org/10.5281/zenodo.3957489.

Abbott, Sam, Joel Hellewell, Robin N Thompson, Katharine Sherratt, Hamish P Gibbs, Nikos I Bosse, James D Munday, et al. 2020. "Estimating the Time-Varying Reproduction Number of SARS-CoV-2 Using National and Subnational Case Counts." *Wellcome Open Res.* 5 (December): 112. https://doi.org/10.12688/wellcomeopenres.16006.2.

Bastos, Leonardo S, Theodoros Economou, Marcelo F C Gomes, Daniel A M Villela, Flavio C Coelho, Oswaldo G Cruz, Oliver Stoner, Trevor Bailey, and Claudia T Codeço. 2019. "A Modelling Approach for Correcting Reporting Delays in Disease Surveillance Data." *Statistics in Medicine* 38 (22): 4363–77. https://doi.org/10.1002/sim.8303.

Bhatt, Samir, Neil Ferguson, Seth Flaxman, Axel Gandy, Swapnil Mishra, and James A Scott. 2020. "Semi-Mechanistic Bayesian Modeling of COVID-19 with Renewal Processes," December. https://arxiv.org/abs/2012.00394.

Bosse, Nikos. 2020. *Scoringutils: A Collection of Proper Scoring Rules and Metrics to Assess Predictions*. https://github.com/epiforecasts/scoringutils.

Champredon, David, Jonathan Dushoff, and David J D Earn. 2018. "Equivalence of the Erlang-Distributed SEIR Epidemic Model and the Renewal Equation." *SIAM J. Appl. Math.* 78 (6): 3258–78. https://doi.org/10.1137/18M1186411.

Cori, Anne, Neil M Ferguson, Christophe Fraser, and Simon Cauchemez. 2013. "A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics." *Am. J. Epidemiol.* 178 (9): 1505–12. https://doi.org/10.1093/aje/kwt133.

Fraser, Christophe. 2007. "Estimating Individual and Household Reproduction Numbers in an Emerging Epidemic." *PLoS One* 2 (8): e758. https://doi.org/10.1371/journal.pone.0000758.

Gabry, Jonah, and Rok Češnovar. 2021. *Cmdstanr: R Interface to 'CmdStan'*.

Günther, Felix, Andreas Bender, Katharina Katz, Helmut Küchenhoff, and Michael Höhle. 2021. "Nowcasting the COVID-19 Pandemic in Bavaria." *Biometrical Journal* 63 (3): 490–502. https://doi.org/10.1002/bimj.202000112.

Höhle, Michael, and Matthias an der Heiden. 2014. "Bayesian Nowcasting During the STEC O104:H4 Outbreak in Germany, 2011." *Biometrics* 70 (4): 993–1002. https://doi.org/10.1111/biom.12194.

Lison, Adrian. n.d. "Nowcast-Transmission." Github.

Meyer, Sebastian, Leonhard Held, and Michael Höhle. 2017. "Spatio-Temporal Analysis of Epidemic Phenomena Using the R Package surveillance." *Journal of Statistical Software* 77 (11): 1–55. https://doi.org/10.18637/jss.v077.i11.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Scott, James A., Axel Gandy, Swapnil Mishra, Juliette Unwin, Seth Flaxman, and Samir Bhatt. 2020. "Epidemia: Modeling of Epidemics Using Hierarchical Bayesian Models."

https://imperialcollegelondon.github.io/epidemia/.
Team, Stan Development. 2020. *Prior Choice Recommendations.*
———. 2021. *Stan Modeling Language Users Guide and Reference Manual, 2.28.1.*