Surveillance model and performance indicator definition

Sam Abbott

In this section, we define the model used as part of our surveillance framework and the approach taken to use this model to evaluate if the underlying relationship between primary and secondary indicators (for example test positive cases and deaths) has recently changed compared to a baseline period. Note that the model we define here has been used successfully, via its implementation in the `EpiNow2` R package[1], to forecast local area hospital admissions[2], to forecast COVID-19 notified deaths in Germany and Poland[3], to forecast COVID-19 notified deaths and hosptial admissions as part of the ECDC forecasting hub project[4], and to forecast notified COVID-19 deaths as part of the CDC forecasting hub project[5], and to provide situational awareness to the Scientific Pandemic Influenza Group on Modelling in the UK[6]. Refer to these references for a detailed evaluation of its use for out of sample forecasting.

Model definition

As the key role of this model framework is to assess if the relationship between primary and secondary notifcations is changing over time we consider a simple model in which a mechanistic relationship is explicitly codified. This model is based on a discrete convolution of primary cases, scaled based on the fraction (here described as the secondary fraction but depending on application potentially being the case fataility ratio, case hospitalisation ratio, or the hospitalisation fatality ratio), and a delay distribution that is assumed to follow a discretised daily log normal distribution. This model can be thought of as a disrete time ordinary differential equation generalised to log normal, rather than exponential, delay distributions. We generalise this simple model beyond the incidence case as described above to also include prevalence indicators (for example hospital admissions and occupancy) where the secondary notifications can be thought of as depending on secondary notifications from the previous timestep, scaled current primary notications, and minus scaled historic primary notifications weighted by some delay distribution. This model can be defined as follows,

$$\hat{S}_t = \delta_p S_t + \alpha \left(\delta_p P_t + \delta_c \sum_{\tau=0}^{D} \xi(\tau|\mu,\sigma) P_{t-\tau} \right) \tag{1}$$

Where S_t and P_t are observed primary and secondary notifications, $\hat{S}_t$ are expected secondary notifications, $\delta_p = 1$ and $\delta_c = -1$ when S_t is a prevalence measure, $delta_p = 0$ and $\delta_c = 1$ when it is an incidence based measure. α and ξ are defined as the secondary fraction and delay from primary to secondary notification (or delay from secondary notification to recovery etc in the prevalence case) with α typically being of most interest to those interpreting the models posterior estimates. We further assume that ξ follows a discretised log normal distibution described by its mean μ and standard deviation σ on the log scale (where we take the cumulative mass function for time t minus the cumlative mass function for $t-1$) normalised by the maximum allowed delay D such that $\sum_{\tau=0}^{D} \xi(\tau|\mu,\sigma) = 1$.

The above definition captures our mechanistic assumptions for the expectation of secondary notifications but does not account for potential observation noise or reporting patterns. Here we assume a negative binomial observation model (though our implementation also supports a Poisson observation model) in order to capture potential reporting overdispersion ϕ and adjust expected counts using an optional day of the week effect based on a simplex $\omega_{(t \mod 7)}$ (such that $\sum_{t=0}^{6} w_t = 7$ so the total effect over a week is balanced). This gives the following observation process.

$$S_t \sim \text{NB}\left(\omega_{t \mod 7} \hat{S}_t, \phi\right) \tag{2}$$

Model priors

We define weakly informative default priors for all parameters based on subject area knowledge. These are as follows,

$$\begin{aligned}
\frac{\omega_{t \mod 7}}{7} &\sim \text{Dirichlet}(1, 1, 1, 1, 1, 1, 1) && (3)\\
\alpha &\sim \mathcal{N}(0.5, 0.25) && (4)\\
\mu &\sim \mathcal{N}(2.19, 0.5) && (5)\\
\sigma &\sim \mathcal{N}(0.47, 0.25) && (6)\\
\phi &\sim \frac{1}{\sqrt{\mathcal{N}(0, 1)}} && (7)
\end{aligned}$$

Note that we choose μ and σ such that the discretised log normal distribution has a mean of 10 and a standard deviation of 5 on the natural scale. We also set maximum delay D to 30 for computational tractibility. Finally, α, σ, and ϕ are truncated to be greater than 0.

Defining a surveillance indicator

We now use the convolution model described in the last section to construct an indicator for whether or not the relationship between primary and secondary notifcations is changing over time. We do this using a counterfactual approach by first fitting the model to a baseline period of time and then projecting this forward to the time period of interest. Under an assumption of no change this approach should provide reasonable predictive accuracy and if it does not we may be able to conclude that there has been an underlying change in the relationship between the two indicators of interest. In order to account for the expected in sample performance of our model (potential misspecification etc.) we normalise performance of predictions from our baseline time period using predictions from a model fit to the time period of interest. This then provides a relative metric of how well the model that assumes no change in indicator relationship captures the current relationship. More detail is provided in the following sections.

Baseline time window versus the time window of interest

We first fit a model to an initial time window which is by default from $w_b + w_t$ (where w_b is the length of the baseline window and w_t is the length of the target window of interest) days before the target date of interest until w_t days before the target date of interest. Optionally we allow for this definition to be varied with overlapping and non-contiguous windows also supported. We then fit another model to the target window of interest (i.e w_t days before the target date). By default we make use of the posterior from the baseline window as a prior for the secondary fraction, delay distribution summary parameters, and if present the overdispersion of the observation model by assuming that the posterior for each parameter is independent normal. This approach allows us to fit to a shorter window of observations and captures our assumption that the baseline window should be representative of the target window. Our implementation also supports inflating these posterior estimates by a multiplicative factor in order to capture our a priori expectation of change in the indicator relationship but our default case is to assume no inflation. This option is potentially of use when the baseline window is thought to be less representative of the target window.

For both models we then generate posterior predictions for the target window of interest (w_t). For the baseline model these represent out of sample predictions and for the target model these represent posterior predictions on observed data. Visualising these predictions allows us to qualitively assess the evidence for the target window being different to the baseline window accounting for model misspecification (as we compare model predictions with and without the use of the observations in the window of interest). We also summarise posterior estimates for secondary fraction, and delay distribution summary parameters for both windows to allow exploration of which aspects of the assumed generative process may differ between the two windows if evidence for this is found when comparing predictions.

Prediction evaluation

In order to evaluate quantitatively if the relationship between indicators is changing between the baseline and target windows we need some measure of quantitative predictive performance. Following recognised good practice, we want this measure to be a proper scoring rules such that the highest expected reward is given if the true probability distribution is supplied as the forecast. Here we make use of the continuous ranked probability score (CRPS) which is a proper scoring rule that generalises the absolute error to probablistic forecasts[7], effectively measuring the 'distance' of the predictive distribution to the observed data-generating distribution[8]. It can be defined as follows,

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

Where y is the true observed value and F the cumulative density function of the predictive distribution. $\mathcal{H}$ is the Heaviside step function and is defined such that it attains the value of 1 if the argument is positive or zero, and the value of 0 otherwise.

Key features of the CRPS that make it attractive for our usage include the fact that it does not distinguish between over- and underconfidence, and it is relatively lenient (when compared to other proper scoring rules) to outlier predictions as it scales linearly due to its relationship to the absolute error[9]. The CRPS is also a global proper scoring rule, vs a local one, meaning that it is sensitive to the distance of the entire predictive distribution from observed value which has been suggested as a desirable property when used in a decision making context[8].

Defining a relative change decision metric

We calculate the CRPS for predictions from both the baseline and target model on the target window of interest and then take the mean for each model. This gives us an absolute measure of performance for any given window and allows us to quantitatively compare models within that window. However, as the CRPS is an absolute measure that depends on the magnitude of secondary observations this does not allow us to compare across target windows (i.e over time) which is a neccessary feature of an indicator designed to be used to signal potential changes in behaviour. In order to produce such a metric we normalise the CRPS for predictions from the target window model using the CRPS for predictions on the target window from the baseline model.

$$\text{Relative change decision metric} = \frac{\text{CRPS for the model fit to target window in the target window}}{\text{CRPS for the model fit to baseline window in the target window}}$$

This produces a relative measure of divergance from baseline performance with no change indicated by values close to 1 and high levels of change being indicated by values closer to 0 (i.e the model fit to the target window starts to outperform the model that is not fit to this data) and accounts for general model misspecification by conditioning expected performance on that idealised performance achieved using data from the target window.

Implementation

The model is implemented using the `EpiNow2` R package (version 1.3.3)[1] `estimate_secondary` convolution model which itself is implemented using `stan` via the `rstan` R package (version 2.21.5)[10,11] and `forecast_secondary` forecasting model which again is also implemented using `stan` and `rstan`. Additional functions from the `EpiNow2` R package are used to facilitate posterior manipulation as well as functions written for this specific implementation and prodived in the code repository. CRPS prediction scoring is implemented using the `scoringRules` R package (version 1.0.1)[12] via the `scoringutils` R package (version 1.0.0)[8]. All data manipulation and pipeline code is written in `R`[13] using `optparse`[14] to create a command line tool.Code for the method outlined in this section can be found here: https://github.com/SACEMA/severity-monitoring

References

1. Abbott, S., Hellewell, J., Sherratt, K., Gostic, K., Hickson, J., Badr, H. S., DeWitt, M., Thompson, R., EpiForecasts, & Funk, S. (2020). *EpiNow2: Estimate real-time case counts and time-varying epidemiological parameters*. https://doi.org/10.5281/zenodo.3957489
2. Meakin, S., Abbott, S., Bosse, N., Munday, J., Gruson, H., Hellewell, J., Sherratt, K., CMMID COVID-19 Working Group, & Funk, S. (2022). Comparative assessment of methods for short-term forecasts of COVID-19 hospital admissions in england at the local level. *BMC Med.*, *20*(1), 86. https://doi.org/10.1186/s12916-022-02271-x
3. Bosse, N. I., Abbott, S., Bracher, J., Hain, H., Quilty, B. J., Jit, M., Centre for the Mathematical Modelling of Infectious Diseases COVID-19 Working Group, Leeuwen, E. van, Cori, A., & Funk, S. (2021). Comparing human and model-based forecasts of COVID-19 in germany and poland. *medRxiv*, 2021.12.01.21266598. https://doi.org/10.1101/2021.12.01.21266598
4. *ECDC forecasting hub*. (n.d.). https://covid19forecasthub.eu/.
5. Cramer, E. Y., Huang, Y., Wang, Y., Ray, E. L., Cornell, M., Bracher, J., Brennen, A., Castro Rivadeneira, A. J., Gerding, A., House, K., Jayawardena, D., Kanji, A. H., Khandelwal, A., Le, K., Niemi, J., Stark, A., Shah, A., Wattanachit, N., Zorn, M. W., ... Consortium, U. C. F. H. (2021). The united states COVID-19 forecast hub dataset. *medRxiv*. https://doi.org/10.1101/2021.11.04.21265886
6. *covid19-uk-nowcasts-projections: Nowcasts and projections of covid-19 in the UK*. (n.d.). Github.
7. Gneiting, T., & Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, *102*(477), 359–378. https://doi.org/10.1198/016214506000001437
8. Bosse, N. I., Gruson, H., Funk, S., EpiForecasts, & Abbott, S. (2020). *Scoringutils: Utilities for scoring and assessing predictions*. https://doi.org/10.5281/zenodo.4618017
9. Machete, R. L. (2012). Contrasting Probabilistic Scoring Rules. *arXiv:1112.4530 [Math, Stat]*. https://arxiv.org/abs/1112.4530
10. Team, S. D. (2021). *Stan modeling language users guide and reference manual, 2.28.1*.
11. Stan Development Team. (2022). *RStan: The R interface to Stan*. https://mc-stan.org/
12. Jordan, A., Krüger, F., & Lerch, S. (2019). Evaluating probabilistic forecasts with scoringRules. *Journal of Statistical Software*, *90*(12), 1–37. https://doi.org/10.18637/jss.v090.i12
13. R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/
14. Davis, T. L. (2021). *Optparse: Command line option parser*. https://CRAN.R-project.org/package=optparse