Supplement: A baseline method for nowcasting count data

2025-04-16

# Supplemental methods

## Mathematical model

This model was initially developed as a reference model for the COVID-19 hospitalisations nowcast challenge in Germany in 2021 and 2022 Wolffram et al. (2023). It uses preliminary case count and their delays in the form of reporting triangles, and uses empirical delay distributions to estimate yet-to-be-observed cases. Probabilistic nowcasts are generated using a negative binomial model with means from the point nowcast and dispersion parameters estimated from past nowcast errors. Below, we describe the mathematical details of each component of the model, starting with a definition of the notation used throughout.

### Notation

We denote as the number of cases occurring on time which appear in the dataset with a delay of . For example, a delay means that a case occurring on day arrived in the dataset on day , or on what is considered to be the first possible report date in practice. We only consider cases reporting within a maximum delay . The number of cases reporting for time with a delay of at most can be written as:

Such that is the “final” number of reported cases on time . Conversely, for

is the number of cases still missing after delay. We refer to to describe a random variable, for the corresponding observation, and for an estimated/imputed value. The matrix of available at a given time is referred to as a reporting matrix. In the case where all have yet to be observed (e.g.  is the current time), this reporting matrix is referred to as the reporting triangle, with all values in the bottom right corner of the triangle being missing, except for the first entry at .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### Delay distribution estimation

#### Estimate of the delay distribution from a reporting matrix

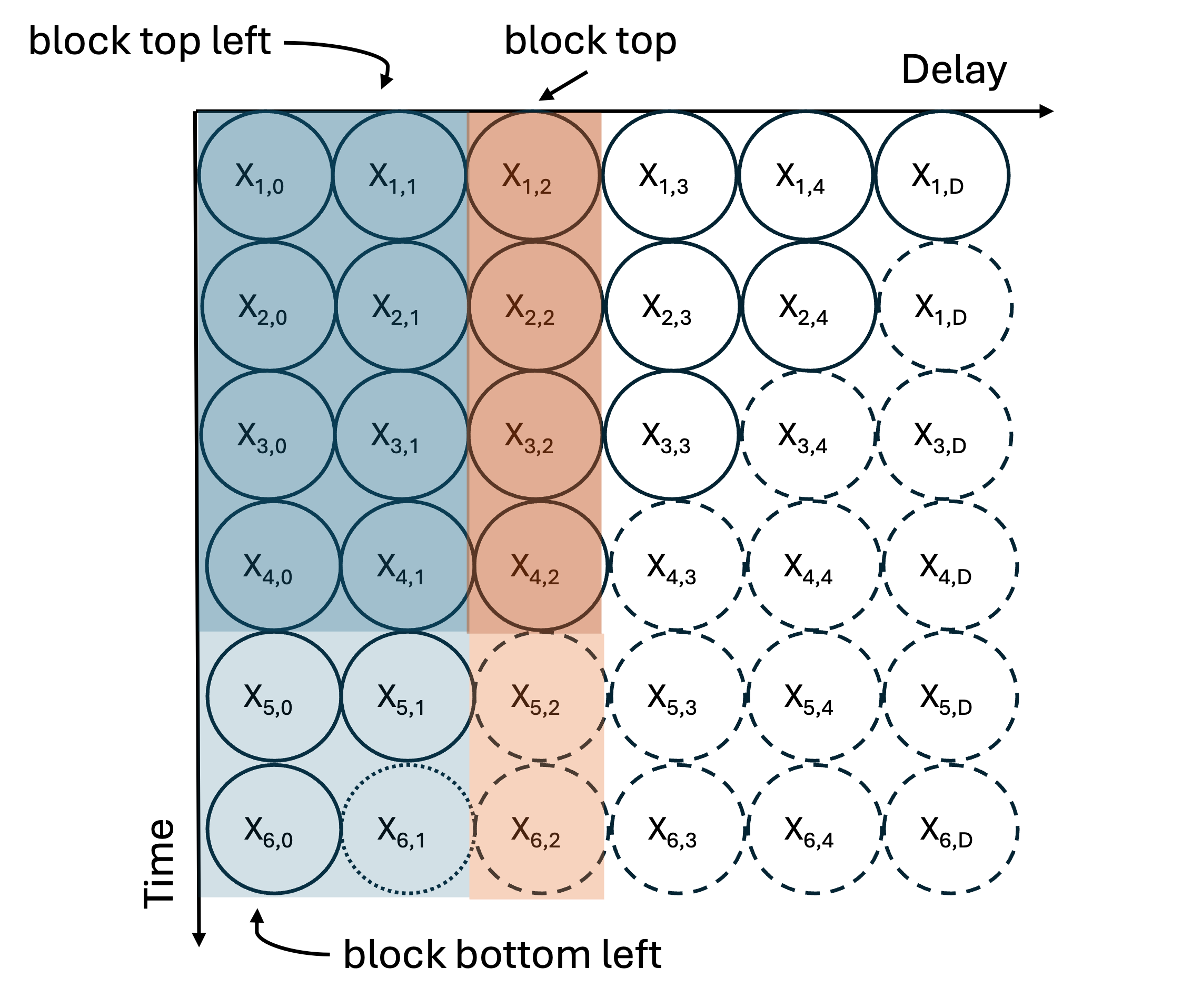
We can use a reporting matrix to compute an empirical estimate of the delay distribution, . The empirical delay distribution, can be computed directly from the reporting matrix

Where the numerator is the sum of all the observations across reference times for a particular delay , and the denominator is the sum across all reference times and delays .

#### Estimate of the delay distribution from a reporting triangle

In the case where we have missing values in the bottom right (i.e. we have a reporting triangle), we need to use the multiplicative model to generate a point nowcast matrix containing a mixture of observed and imputed values. Then we can compute the delay distribution as described above for the reporting matrix case.

The multiplicative model works by iteratively “filling in” the reporting triangle starting from the bottom left, and moving column by column from left to right until the bottom right of the triangle is filled in.



Visual description of the iterative “completing” of the reporting triangle, moving from left to right and bottom to top. In this cases, we are imputing $x\_{t=6, d = 2}$ and $x\_{t=5, d= 2}$ assuming that the ratio between $x\_{t=1:4, d = 2}$ (block top), and $x\_{t=1:4, d=0:1}$ (block top left) holds for for $x\_{t=5:6, d = 2}$ (block bottom) and $x\_{t=5:6, d = 0:1}$ (block bottom left). In this example, $\hat{x}\_{t=6, d = 1}$ has already been imputed using the same approach, and we treat it as known going forward. This process is repeated across the reporting triangle to estimate all values outlined in the dashed lines.

The method requires at least one observation, at delay for the most recent reference time, located at the bottom left of the reporting triangle in Figure @ref(fig1) above. This means that in practice, the reporting triangle must have at least the same number of rows as it does columns. The method assumes that the values at each delay for the recent times, , will consist of the same proportion of the values previously reported for earlier times . To fill in the missing values for each column , we sum over the rectangle of completely observed reference times for all columns (block top left) and sum over the column of completely observed reference times for all of the entries in column (block left). The ratio of these two sums is assumed to be the same in the missing entries in column , so we use the entries observed up to for each incomplete reference time (block bottom left), and scale by this ratio to get the missing entries in column . This process is repeated for each delay from up to the maximum delay . At each iteration an additional reference time entry is computed as the delay increases.

The delay distribution is then estimated from the filled in reporting matrix, using the same algorithm as described above for the case of the complete reporting matrix.

### Point nowcast generation

To “fill in” the reporting triangle from the delay distribution, we need to estimate the expected total number of eventual observed cases , for each reference time . Let be the sum over all delays that have already been observed (up until ), such that and be the cumulative sum of the delay distribution, up until such that . By assuming that and , it can be shown (See [Zero-handling](#zero-handling-approximation) below) that the expected value of , the total number of reported cases on reference time , can be written as:

Then we can compute directly using the th element of

Where the number of reports at timepoint with delay is the product of the the expected total reports, and the proportion expected at that particular delay , .

### Uncertainty estimation

To estimate the uncertainty in the nowcasts, we use past nowcast errors and assume a negative binomial observation model.

#### Generation of retrospective reporting triangles

We describe a method which generates retrospective reporting triangles to replicate what would have been available as of time , where for to generate retrospective reporting triangles.

To generate the set of reporting triangles, we simply remove the last rows of the existing reporting triangle (or reporting matrix), to generate truncated incomplete reporting matrices. The method uses each retrospective reporting triangle to re-estimate a delay distribution using the preceding rows of the reporting triangle before , and recomputes a retrospective nowcast, for realizations of the retrospective reporting triangle (so different values).

#### Generation of retrospective point nowcast matrices

From the reporting triangles, we apply the method described above to estimate a delay distribution from a reporting triangle and generate a point nowcast for each reporting triangle, to generate point nowcasts.

#### Fit point nowcast matrices and observed values to a negative binomial observation model at each delay

We then take the list of point nowcast matrices, and the list of truncated incomplete reporting matrices (these do not necessarily contains NAs on the bottom right, the bottom right could be entirely or partially observed). For each delay we identify the overlapping set of matrix elements that were imputed retrospectively and matrix elements that had been observed as of the most recent time point.

For each delay we assume that the observed values, follow a negative binomial observation model with a mean of :

We add a small number (0.1) to the mean to avoid an ill-defined negative binomial. We note that to perform all these computations, data snapshots from at least past observations, or rows of the original reporting triangle (or matrix), are needed. This estimate of the uncertainty accounts for the empirical uncertainty in the point estimate of the delay distribution over time.

### Probabilistic nowcast generation

Using the dispersion parameters for each delay, for , we can generate probabilistic nowcast matrices by drawing samples from the negative binomial:

### Nowcast creation

To generate nowcast vectors that represent the expected final number of cases at each reference time, we aggregate the reporting matrices which contain a mix of observed and imputed values. We refer to the sum across delays as , or

We can then used the probabilistic draws to compute any desired quantiles to summarize the outputs.

### Zero-handling approximation

\*\* To be filled in\*\*

## Additional figures

### German nowcast Hub validation

1. Relative WIS by age group
2. Mean WIS by nowcast horizon for each model—same panel relative WIS by nowcast horizon.
3. Empirical coverage at 50% and 90% prediction intervals for each model.
4. Empirical coverage at 50% and 90% prediction intervals by nowcast horizon for each model.
5. Empirical coverage at 50% and 90% prediction intervals by age group for each model configuration.

### Model permutation study

1. Absolute WIS by nowcast horizon for each model configuration. Colour by model configuration, decomposed by dispersion, overprediction, and underprediction.
2. Absolute WIS by age group for each model configuration. Colour by model configuration, decomposed by dispersion, overprediction, and underprediction.
3. Absolute WIS over time (by week) for each model configuration. Colour by model configuration, decomposed by dispersion, overprediction, and underprediction.

### Case study: UKHSA norovirus surveillance

1. Absolute WIS over time (by week) for each model.
2. Absolute WIS by day of week for each model.
3. Empirical coverage at 50% and 90% prediction intervals for each model
4. Empirical coverage at 50% and 90% prediction intervals by day of week for each model.

## References

Wolffram, Daniel, Sam Abbott, Matthias an der Heiden, Sebastian Funk, Felix Günther, Davide Hailer, Stefan Heyder, et al. 2023. “Collaborative Nowcasting of COVID-19 Hospitalization Incidences in Germany.” *PLOS Computational Biology* 19 (8): 1–25. <https://doi.org/10.1371/journal.pcbi.1011394>.