All model runner options

Source: `vignettes/all-model-options.Rmd`

This article breaks down all the options available when running `mbg` models. For a summary of these terms, see the documentation for the mbg::MBGModelRunner$new() method

1: MBG model basics

The model always requires two terms: `input_data`, which includes all point observations of the outcome to be estimated, and `id_raster`, which lays out the study area.

1.A: Input data

Formatted as a `data.frame` or `data.table::data.table`. Should contain at least the following fields:

- `indicator`: Contains the outcome to be modeled. For binomial or Poission MBG models, this is the numerator. For Gaussian models, this is the observed mean.
- `samplesize`: Only required for binomial or Poisson data. This is the denominator for each observations.
- `sd`: Only required for Gaussian models. This is the observed standard deviation for each observation.
- `x`: The point's x position, in the same coordinate reference system as the ID raster. For unprojected data, this is the longitude
- `y`: The points y position, in the same coordinate reference system as the ID raster. For unprojected data, this is the latitude
- `cluster_id`: A unique identifier for each row in the data

1.B: ID raster

A `terra::SpatRaster` object meeting the following requirements:

- Has a single layer
- Covers the entire study area

- Contains non-NA pixel values for pixels that should be estimated by the model

An ID raster can be created using the `mbg::build_id_raster` function.

Before running a model, you could use the `terra::extract` function to ensure that all points in your input data overlap a non-NA pixel in the ID raster.

2: Model family and link function

The arguments `inla_family`, `inla_link`, and `inverse_link` give the relationship between the observed data and the linear combination of effects that make up the model. The model defaults specify a binomial likelihood:

- `inla_family = 'binomial'`
- `inla_link = 'logit'`
- `inverse_link = 'plogis'`

For binomial data, each data point with numerator y_i and denominator N_i is evaluated against a probability p_i, which is governed by a logit-linear combination of model effects:

$$
\begin{aligned}
y_i &\sim Binomial(N_i, p_i) \\
logit(p_i) &= \; ...
\end{aligned}
$$

The actual model effects ($...$) are described in the following section.

3: Model effects

The model currently has four effect types which can be toggled and controlled via settings passed to `mbg::MbgModelRunner`.

3.A: Covariate effects

Relevant settings:

- `use_covariates` (default `TRUE`)
- `covariate_rasters` (default `NULL`)
- `use_stacking` (default `FALSE`)

A covariate effect will only be included if `use_covariates` is `TRUE` (the default) and `covariate_rasters` are passed. The `covariate_rasters` are an optional list of `terra::SpatRaster` pixel-level predictive covariates. They can be incorporated into the model in two different ways depending on the value of `use_stacking`:

3.A.i: Standard covariate effect

Only applied if a covariate effect is included and `use_stacking` is `FALSE` (the default).

The covariate effect at observation i is $\gamma_i^{covariates} = \vec{\beta} X_{s_i}$, where $\vec{\beta}$ are linear effects on the matrix of covariate values X evaluated at the location of observation i (s_i).

Note that an intercept is not included by default. If you want a model with no covariate effects other than an intercept, pass a `covariate_rasters` with an `intercept` raster containing all 1s.

A prior is applied to the variance of effects on all covariates other than the intercept: `prior_covariate_effect` (default `list(threshold = 3, prob_above = 0.05)`) is a penalized complexity prior that can be expressed as a level of certainty about the standard deviation on each fixed effect β. For example, the default prior corresponds to $P(\sigma_\beta > 3) = 0.05$.

3.A.ii: Stacked ensemble model

Only applied if a covariate effect is included and `use_stacking` is `TRUE`.

For a stacked ensemble model, the covariate effect for observation i is:

$$\gamma_i^{covariates} = \sum_{j=1}^{J} [w_j f_j(X_{s_i})]$$
$$Constraints: w_j > 0 \, \forall \, j, \; \sum_{j=1}^{J}(w_j) = 1$$

Where:

- $\vec{f}$: Predictions from a set of J regression models fit to the raw covariate data X
- $\vec{w}$: A weighting vector corresponding to each regression model f_j. The weights are constrained to be strictly greater than zero and to sum to one.
- X_{s_i}: The raw covariate values at the location of observation i, s_i

Relevant model settings:

- `stacking_model_settings` (default `list(gbm = NULL, treebag = NULL, rf = NULL)`): Defines the list of component models $f_j(X)$ to be fitted to the covariates. A named list—each name corresponds to a regression model in the `caret` package, and each value stores optional settings that can be passed to that model.

- `stacking_cv_settings` (default `list(method = 'repeatedcv', number = 5, repeats = 5)`): These are used by `caret::traincontrol` to cross-validate each regression model
- `stacking_use_admin_bounds` (default `FALSE`), `admin_bounds` (default `NULL`), `admin_bounds_id` (default `NULL`): If `stacking_use_admin_bounds` is `TRUE` and the other two values are set, adds administrative fixed effects to each of the component models.
- `stacking_prediction_range` (default `NULL`): Can be used to restrict the prediction range of each component regression model. For binomial data, a reasonable limit is to not predict outside of `c(0, 1)`.

3.B: Gaussian process

If the setting `use_gp` is `TRUE` (the default), adds a spatially correlated effect:

$$Z \sim GP(0, \Sigma_s)$$

Where Z is a Gaussian process with mean zero and stationary isotropic Matern covariance over space (Σ_s).

The Gaussian process is informed by priors on the range and variance:

- `prior_spde_range` (default `list(threshold = 0.1, prob_below = 0.05)`) Prior on the geostatistical range of the Gaussian process, the distance beyond which there is little or no spatial autocorrelation between pairs of points on the GP. This is a penalized complexity prior expressed as a named list with two items. The `threshold` is a distance expressed *relative to the diameter of the study area*: for example, the default `threshold` of 0.1 corresponds to a geostatistical range equivalent to one-tenth the diameter of the study area. The `prob_below` is the probability that the true range falls below this threshold. In other words, the default prior is $P(range < \frac{diameter}{10}) = 0.05$
- `prior_spde_sigma` (default `list(threshold = 3, prob_above = 0.05)`) Penalized complexity prior on the variance of the Gaussian process, expressed in terms of the standard deviation σ_Z. The default corresponds to a prior belief that $P(\sigma_Z > 3) = 0.05$

To simplify estimation, the R-INLA package represents the continuous Gaussian process on a 2D spatial mesh. Three more settings control the mesh:

- `mesh_max_edge` (default `c(0.2, 5.0)`): The maximum length of a mesh edge in areas where there is little to no data. Expressed in the same units of measurement as the projection used for the `id_raster` (for unprojected data, this is decimal degrees). The first term is the maximum edge length within the study area, and the second term is the maximum edge length outside the study area (the mesh extends beyond the study area to mitigate edge effects).

- `mesh_cutoff` (default `0.04`): The minimum length of a mesh edge in areas where data is dense. Expressed in the same units of measurement as `mesh_max_edge`.
- `spde_integrate_to_zero` (default `FALSE`): Should the volume under the fitted mesh integrate to zero?

For more details about the INLA approach to approximate Gaussian process regression, see the papers at the bottom of this page.

3.C: Administrative-level effect

This effect is a random intercept grouped by administrative unit. The administrative level (polygon boundaries) of interest can be set by the user. If the effect is on, then the following term is added:

$$\gamma_{a_i}^{admin} \sim N(0, \sigma^2_{admin})$$

In other words, $\vec{\gamma}^{admin}$ is an vector of random intercepts with length equal to the total number of administrative units, IID normal with mean 0 and variance σ^2_{admin}. All observations i in the same administrative division a share the same intercept $\gamma_{a_i}^{admin}$.

Relevant settings:

- `use_admin_effect` (default `FALSE`): Should the administrative-level effect be included in the model?
- `prior_admin_effect` (default `list(threshold = 3, prob_above = 0.05)`): A prior applied to the administrative effect variance, expressed in terms of the standard deviation. The default settings correspond to the prior belief that $P(\sigma_{admin} > 3) = 0.05$
- `admin_bounds` (default `NULL`): Administrative bounds that will be used to group observations.
- `admin_bounds_id` (default `NULL`): Unique identifier field for `admin_bounds`

3.D: Nugget

The nugget is an independently and identically distributed (IID) normal effect applied to each observation. It corresponds to “irreducible variation” not captured by any other model effect:

$$\gamma_i^{Nugget} \sim N(0, \sigma^2_{nugget})$$

Relevant settings:

- `use_nugget` (default `TRUE`): Should the nugget effect be included in model fitting?
- `prior_nugget` (default `list(threshold = 3, prob_above = 0.05)`): A prior applied to the nugget variance, expressed in terms of the standard deviation. The default settings correspond to the prior belief that $P(\sigma_{nugget} > 3) = 0.05$
- `nugget_in_predict` (default `TRUE`): If `TRUE`, independent samples from $N(0, \sigma^2_{nugget})$ are added to each pixel-level predictive draw.

4: Aggregation to polygon boundaries

As shown in the introductory tutorial, the `mbg::MbgModelRunner` object can automatically aggregate predictions to administrative boundaries. The following three objects are required to perform aggregation:

- `aggregation_table`: A table created by `mbg::build_aggregation_table`. Contains information about the proportional area of each pixel within each administrative boundary polygon.
- `aggregation_levels`: A named list, where each name corresponds to the administrative aggregation level, and each value is a character vector with corresponding grouping fields in the `aggregation_table`.
- `population_raster`: A raster with the same dimensions as `id_raster` that contains population estimates for each pixel. Aggregation from pixels to administrative boundaries accounts for varying pixel-level populations as well as fractional pixel areas.

5: Logging

Finally, the setting `verbose` (default `TRUE`) governs whether the model will perform detailed logging. You can access model logs afterwards by running `mbg::logging_get_timer_log`.

6: Further reading

Bakka, H., *et al.* (2018). Spatial modeling with R-INLA: A review. Wiley Interdisciplinary Reviews: Computational Statistics, 10(6), e1443. https://doi.org/10.1002/wics.1443

Bhatt, S., Cameron, E., Flaxman, S. R., Weiss, D. J., Smith, D. L., & Gething, P. W. (2017). *Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalization.*

Journal of The Royal Society Interface, 14(134), 20170520. https://doi.org/10.1098/rsif.2017.0520

Freeman, M. (2017). *An introduction to hierarchical modeling*. http://mfviz.com/hierarchical-models/

Moraga, Paula. (2019). Geospatial Health Data: Modeling and Visualization with R-INLA and Shiny. Chapman & Hall/CRC Biostatistics Series. ISBN 9780367357955. https://www.paulamoraga.com/book-geospatial/index.html

Opitz, T. (2017). *Latent Gaussian modeling and INLA: A review with focus on space-time applications*. Journal de la société française de statistique, 158(3), 62-85. https://www.numdam.org/article/JSFS_2017__158_3_62_0.pdf

Developed by Nathaniel Henry, Benjamin Mayala.

Site built with pkgdown 2.1.3.