

Package: bmetapred (via r-universe)

October 29, 2026

Title Bayesian Meta-Prediction in the Presence of Prediction Error

Version 0.1.0

Description Fits pseudo-Bayesian Model Averaging (pBMA) and Bayesian Stacking ensembles of linear meta-models while explicitly accounting for the prediction errors of the individual models, building on the Bayesian Stacking of predictive distributions framework of Yao et al. (2018) <[doi:10.1214/17-BA1091](https://doi.org/10.1214/17-BA1091)>. When native uncertainty estimates are unavailable, it provides tools to estimate heteroscedastic errors via Local Polynomial Regression (LOESS). The meta-model space is explored using exhaustive search, a Markov Chain Monte Carlo Model Composition (MC3) algorithm following Madigan and Raftery (1994) <[doi:10.1080/01621459.1994.10476894](https://doi.org/10.1080/01621459.1994.10476894)> when pBMA is selected, or a tabu-guided heuristic search with parallel tempering when bayesian stacking is selected, all powered by 'brms' (Bürkner, 2017 <[doi:10.18637/jss.v080.i01](https://doi.org/10.18637/jss.v080.i01)>).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 8.0.0

Imports brms, future, future.apply, loo, stats, utils

NeedsCompilation no

Author Jorge Piqueras Marqués [aut, cre] (ORCID: <<https://orcid.org/0000-0002-6812-2106>>), Anabel Forte Deltell [aut] (ORCID: <<https://orcid.org/0000-0001-9534-1817>>), Joaquin Martinez Minaya [ctb] (ORCID: <<https://orcid.org/0000-0002-1016-8734>>)

Maintainer Jorge Piqueras Marqués <jorge.piqueras.m@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-10-29 16:17:00 UTC

RemoteUrl <https://github.com/cran/bmetapred>

RemoteRef HEAD

RemoteSha 4a581a7bf789bf7afdfa6106d5d34e2245960a20

Contents

fit_bmetapred	2
fit_LOESS	4
get_pip_bmetapred	5
predict.bmetapred_loess	6
predict.bmetapred_model	7
summary.bmetapred_model	8
Index	10

fit_bmetapred	<i>Fit an Ensemble of bayesian linear meta-models with prediction errors</i>
---------------	--

Description

Fits a Pseudo-Bayesian model averaging (pBMA) or Stacking ensemble of models. It accounts for prediction errors in the individual models. Uses either an exhaustive search or a Markov Chain Monte Carlo (MC3) algorithm to explore the meta-model space and calculate pBMA/Stacking weights.

Usage

```
fit_bmetapred(  
  data_train,  
  y,  
  predictors,  
  sd_cols,  
  method = c("stacking", "pbma"),  
  p_max_exhaust = 9,  
  n_iter_mc3 = 1500,  
  burnin_pbma = NULL,  
  min_K = 10,  
  max_K = 50,  
  pri_sd = 2.5,  
  pri_sd_int = 5,  
  pri_sd_sigma = 0.5,  
  pri_sd_int_sigma = 5,  
  parallel_workers = 1,  
  cores_stan = 1,  
  iter_stan = 1500,  
  verbosity = 1,  
  backend = "rstan"  
)
```

Arguments

data_train	A data frame containing the training data.
y	A character string specifying the name of the target variable.
predictors	A character vector specifying the names of the predictor variables (models).
sd_cols	A character vector specifying the names of the columns containing the standard deviations (prediction errors) for each predictor (model). Must be in the same order as predictors.
method	A character string specifying the ensemble method. Can be either "stacking" (default) or "pbma".
p_max_exhaust	Integer. The maximum number of predictors to perform an exhaustive search. If the number of predictors exceeds this value, the MC3 algorithm is used. Default is 9.
n_iter_mc3	Integer. Number of iterations for the MC3 algorithm. Default is 1500.
burnin_pbma	Integer. Number of burn-in iterations to discard when using the "pbma" method. If NULL, it defaults to 20% of n_iter_mc3.
min_K	Integer. Minimum number of top models to evaluate before early stopping in the Stacking matrix compilation. Default is 10.
max_K	Integer. Maximum number of top models to evaluate in the Stacking matrix. Default is 50.
pri_sd	Numeric. Prior standard deviation for the regression coefficients. Default is 2.5.
pri_sd_int	Numeric. Prior standard deviation for the intercepts. Default is 5.
pri_sd_sigma	Numeric. Prior standard deviation for the sigma coefficients. Default is 0.5.
pri_sd_int_sigma	Numeric. Prior standard deviation for the sigma intercept. Default is 2.
parallel_workers	Integer. Number of workers for parallel evaluation of the model space. Default is 1 (sequential execution).
cores_stan	Integer. Number of cores to use for each underlying brms model. Default is 1.
iter_stan	Integer. Number of MCMC iterations for the underlying brms models. Default is 1500.
verbosity	Integer. Level of printed messages. 0 suppresses output, 1 prints progress. Default is 1.
backend	Character string specifying the package to use as the backend for brms. Default is "rstan". However, for intensive computational tasks and large model spaces, we highly recommend installing "cmdstanr" (available from the Stan-dev repository) and setting backend = "cmdstanr" to improve memory management and speed.

Value

An object of class `bmetapred_model` containing the optimized ensemble weights, the list of selected model indices, the compiled brms base templates, and internal parameters needed for prediction.

Examples

```
set.seed(123)
df <- data.frame(
  Y = rnorm(30),
  M1 = rnorm(30), M2 = rnorm(30),
  sd_M1 = rep(0.1, 30), sd_M2 = rep(0.2, 30)
)

my_ensemble <- fit_bmetapred(
  data_train = df,
  y = "Y",
  predictors = c("M1", "M2"),
  sd_cols = c("sd_M1", "sd_M2"),
  method = "stacking",
  parallel_workers = 1,
  cores_stan = 1,
  iter_stan = 300,
  verbosity = 0
)
summary(my_ensemble)
```

fit_LOESS

Fit LOESS Error Models for Ensemble Predictors

Description

Trains a Local Polynomial Regression (LOESS) model for each predictor to estimate its prediction error (standard deviation) based on historical validation data. This is particularly useful when base predictors do not output native uncertainty estimates. The LOESS model predicts the squared error as a function of the predicted value.

Usage

```
fit_LOESS(data, y, predictors, span = 0.4)
```

Arguments

data	A data frame containing the historical or validation data.
y	A character string specifying the name of the true target variable.
predictors	A character vector specifying the names of the predictor variables.
span	Numeric. The smoothing parameter for the LOESS model. Default is 0.4.

Value

An object of class `bmetapred_loess` containing the fitted LOESS models and the extrapolation boundaries for each predictor.

Examples

```
set.seed(123)
df_val <- data.frame(Y = rnorm(50), M1 = rnorm(50), M2 = rnorm(50))
loess_estimators <- fit_LOESS(df_val, y = "Y", predictors = c("M1", "M2"))
```

get_pip_bmetapred	<i>Extract Posterior Inclusion Probabilities (PIP)</i>
-------------------	--

Description

Calculates the Posterior Inclusion Probability (PIP) for each predictor (model) variable evaluated during the ensemble fitting process.

Usage

```
get_pip_bmetapred(object)
```

Arguments

object An object of class bmetapred_model returned by fit_bmetapred.

Value

A named numeric vector sorted in decreasing order, representing the cumulative probability (or weight) of inclusion for each base model.

Examples

```
# Create a small bmetapred_model object for fast testing
mock_ensemble <- structure(
  list(
    pesos = c(0.6, 0.3, 0.1),
    top_indices = list(c(1, 1, 0), c(1, 0, 1), c(0, 1, 0)),
    params = list(predictors = c("M1", "M2", "M3")),
    method = "stacking"
  ),
  class = "bmetapred_model"
)
pips <- get_pip_bmetapred(mock_ensemble)
```

predict.bmetapred_loess

Predict Standard Deviations using Fitted LOESS Error Models

Description

Applies fitted LOESS error models to new data to estimate the standard deviation (prediction error) of each predictor. It includes robust handling of extrapolations by capping predictions at the boundary values of the original training data to prevent NAs or collapsed variances.

Usage

```
## S3 method for class 'bmetapred_loess'
predict(object, newdata, prefix = "sd_", min_var = 4e-04, ...)
```

Arguments

object	An object of class bmetapred_loess returned by fit_LOESS.
newdata	A data frame containing the new predictions to evaluate.
prefix	Character. The prefix to add to the predictor names to create the new standard deviation columns. Default is "sd_".
min_var	Numeric. The minimum allowed variance to prevent zero or negative variances (singularities). Default is 0.0004 (SD = 0.02).
...	Additional arguments passed to methods.

Value

A data frame identical to newdata but with appended columns containing the estimated standard deviations for each predictor.

Examples

```
set.seed(123)
df_val <- data.frame(Y = rnorm(50), M1 = rnorm(50), M2 = rnorm(50))
loess_estimators <- fit_LOESS(df_val, y = "Y", predictors = c("M1", "M2"))

df_new <- data.frame(M1 = rnorm(10), M2 = rnorm(10))
df_ready <- predict(loess_estimators, newdata = df_new)
```

predict.bmetapred_model

Predict Method for Ensemble Models

Description

Generates posterior predictive samples and summary statistics for new data based on a fitted ensemble model. It samples from the underlying brms base models proportionally to their optimized weights.

Usage

```
## S3 method for class 'bmetapred_model'
predict(
  object,
  newdata,
  n_samples_totales = 2000,
  estimator = "ensemble",
  occam_window = NULL,
  iter_stan = NULL,
  cores_stan = NULL,
  verbosity = 1,
  ...
)
```

Arguments

object	An object of class bmetapred_model returned by fit_bmetapred.
newdata	A data frame containing the new data to predict. Must include the predictor variables and their corresponding standard deviation columns as defined in the original training phase.
n_samples_totales	Integer. The total number of posterior predictive samples to generate across the ensemble. Default is 2000.
estimator	Character. For "pbma" models, dictates the prediction strategy. Use "ensemble" for weighted averaging or "HPM" (Highest Probability Model) to predict using only the top model. Ignored if the method is "stacking".
occam_window	Numeric. A value between 0 and 1 defining the Occam's window for predictive mass. Models falling outside this cumulative probability are discarded for predicting, and remaining weights are renormalized. Default is NULL.
iter_stan	Integer. Number of MCMC iterations for underlying brms prediction. Defaults to the value used during training.
cores_stan	Integer. Number of cores for underlying brms prediction. Defaults to the value used during training.

verbosity	Integer. Level of printed messages. 0 suppresses output, 1 prints progress. Default is 1.
...	Additional arguments passed to methods.

Value

A data frame containing the predicted estimate (mean), lower_95, and upper_95 credible intervals. If present in newdata, the truth and datetime columns are also appended. The raw predictive matrix is attached as an attribute named "raw_samples".

Examples

```
set.seed(123)
df_train <- data.frame(
  Y = rnorm(30),
  M1 = rnorm(30), M2 = rnorm(30),
  sd_M1 = rep(0.1, 30), sd_M2 = rep(0.2, 30)
)
df_test <- data.frame(
  M1 = rnorm(5), M2 = rnorm(5),
  sd_M1 = rep(0.1, 5), sd_M2 = rep(0.2, 5)
)

my_ensemble <- fit_bmetapred(
  data_train = df_train,
  y = "Y",
  predictors = c("M1", "M2"),
  sd_cols = c("sd_M1", "sd_M2"),
  method = "stacking",
  parallel_workers = 1,
  cores_stan = 1,
  iter_stan = 300,
  verbosity = 0
)

predictions <- predict(my_ensemble, newdata = df_test, n_samples_totales = 200, verbosity = 0)
head(predictions)
```

summary.bmetapred_model

Summarize a bmetapred Ensemble Model

Description

Provides a tabular summary of the meta-models included in the final ensemble, displaying their structural formula, rank, and assigned weight.

Usage

```
## S3 method for class 'bmetapred_model'  
summary(object, ...)
```

Arguments

object	An object of class bmetapred_model returned by fit_bmetapred.
...	Additional arguments passed to methods.

Value

A data frame detailing the rank, weight, and structure of the meta-models that constitute the ensemble. The result is also printed to the console.

Examples

```
# Create a lightweight mock bmetapred_model object for fast testing  
mock_ensemble <- structure(  
  list(  
    pesos = c(0.6, 0.3, 0.1),  
    top_indices = list(c(1, 1, 0), c(1, 0, 1), c(0, 1, 0)),  
    params = list(predictors = c("M1", "M2", "M3")),  
    method = "stacking"  
  ),  
  class = "bmetapred_model"  
)  
summary_table <- summary(mock_ensemble)
```

Index

`fit_bmetapred`, [2](#)

`fit_LOESS`, [4](#)

`get_pip_bmetapred`, [5](#)

`predict.bmetapred_loess`, [6](#)

`predict.bmetapred_model`, [7](#)

`summary.bmetapred_model`, [8](#)