

Package: lopensemble (via r-universe)

June 1, 2026

Title Create Mixture Models From Predictive Samples

Version 0.1.2.9000

Description Combines predictions from individual time series or panel data models into an ensemble using stacking (Yao, Vehtari, Simpson, and Gelman (2018) <doi:10.1214/17-BA1091>) based on the Continuous Ranked Probability Score (CRPS) (Gneiting and Raftery (2007) <doi:10.1198/016214506000001437>) over k-step ahead predictions. Predictions must be predictive distributions represented by samples, typically posterior predictive simulation draws from a Markov chain Monte Carlo (MCMC) algorithm. Given training data with observed values and predictive samples from different models, optimal stacking weights are computed to minimize expected cross-validation predictive error. These weights can then be used to generate samples from the mixture model by drawing from individual model predictions in the correct proportions.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports data.table, cmdstanr

Remotes stan-dev/cmdstanr

SystemRequirements CmdStan (>=2.29)

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), scoringutils

VignetteBuilder knitr

Config/testthat/edition 3

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

Date/Publication 2026-01-30 14:51:24 UTC

RemoteUrl <https://github.com/epiforecasts/lopensemble>

RemoteRef v0.1.3

RemoteSha cd260b8a8dac089bd8d188fc32ef937f988ad296

Contents

crps_weights	2
mixture_from_samples	3
Index	5

crps_weights	<i>Obtain CRPS stacking weights</i>
--------------	-------------------------------------

Description

given true values and predictive samples from different models, ‘crps_weights’ returns the stacking weights which produce the ensemble that minimises the Continuous Ranked Probability Score (CRPS).

Usage

```
crps_weights(data, lambda = NULL, gamma = NULL, dirichlet_alpha = 1.001)
```

Arguments

data	a data.frame with the following entries: <ul style="list-style-type: none"> • observed, the true observed values • predicted, predicted values corresponding to the true values in observed • model, the name of the model used to generate the corresponding predictions • geography (optional), the regions for which predictions are generated. If geography is missing, it will be assumed there are no geographical differences to take into account. Internally, regions will be ordered alphabetically • date (the date of the corresponding prediction / true value). Also works with numbers to indicate timesteps
lambda	weights given to timepoints. If lambda is NULL, the default gives more weight to recent time points with $\text{lambda}[t] = 2 - (1 - t / T)^2$. Note that elements of lambda need not necessarily sum up to one as the stan model automatically constraints the final weights to sum to one irrespective of lambda. lambda = "equal" uses equal weights
gamma	weights given to regions. If gamma is NULL the default is equal weights for the regions. Weights are mapped to regions alphabetically, so make sure that the weights correspond to the regions in alphabetical order.
dirichlet_alpha	prior for the weights. Default is 1.001

Value

returns a vector with the model weights

References

Strictly Proper Scoring Rules, Prediction, and Estimation, Tilmann Gneiting and Adrian E. Raftery, 2007, Journal of the American Statistical Association, Volume 102, 2007 - Issue 477

Using Stacking to Average Bayesian Predictive Distributions, Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman, 2018, Bayesian Analysis 13, Number 3, pp. 917–1003

Examples

```
## Not run:
library("data.table")
splitdate <- as.Date("2020-03-28")
data <- setDT(example_data)

traindata <- data[date <= splitdate]
testdata <- data[date > splitdate]

weights <- crps_weights(traindata)

## End(Not run)
```

mixture_from_samples *Make mixture model from predictive samples*

Description

The function takes a data.frame with predictive samples generated from different models as well as weights corresponding to these models as input. It then returns predictive samples from a mixture model generated by stacking the original models using these weights.

Usage

```
mixture_from_samples(data, weights = NULL, ...)
```

Arguments

data a data.frame with the following entries:

- **observed**, the true observed values (optional)
- **predicted**, predicted values corresponding to the true values in observed
- **model**, the name of the model used to generate the corresponding predictions
- **geography** (optional), the regions for which predictions are generated. If geography is missing, it will be assumed there are no geographical differences to take into account. Internally, regions will be ordered alphabetically

- date (the date of the corresponding prediction / true value). Also works with numbers to indicate timesteps
- weights stacking weights used to combine the original model to a mixture model. If NULL (default), weights will first be estimated using [crps_weights()].
- ... any additional parameters to pass to [crps_weights()] if 'weights' is NULL.

Value

data.frame with samples from the mixture model. The following columns are returned:

- observed, the true observed values, if they were given as input
- predicted, predicted values corresponding to the true values in observed
- model, the name of the model used to generate the corresponding predictions
- geography (optional), the regions for which predictions are generated. If geography is missing, it will be assumed there are no geographical differences to take into account. Internally, regions will be ordered alphabetically
- date (the date of the corresponding prediction / true value). Also works with numbers to indicate timesteps

References

Using Stacking to Average Bayesian Predictive Distributions, Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman, 2018, Bayesian Analysis 13, Number 3, pp. 917–1003

Examples

```
## Not run:
library("data.table")
data <- setDT(example_data)
weights <- c(0.2, 0.3, 0.4, 0.1)
mix <- mixture_from_samples(data, weights = weights)

## End(Not run)
```

Index

`crps_weights`, [2](#)

`mixture_from_samples`, [3](#)