

Package ‘conformalbayes’

July 25, 2025

Title Jackknife(+) Predictive Intervals for Bayesian Models

Version 0.1.4

Description Provides functions to construct finite-sample calibrated predictive intervals for Bayesian models, following the approach in Barber et al. (2021) <[doi:10.1214/20-AOS1965](https://doi.org/10.1214/20-AOS1965)>. These intervals are calculated efficiently using importance sampling for the leave-one-out residuals. By default, the intervals will also reflect the relative uncertainty in the Bayesian model, using the locally-weighted conformal methods of Lei et al. (2018) <[doi:10.1080/01621459.2017.1307116](https://doi.org/10.1080/01621459.2017.1307116)>.

Imports cli, rstantools, loo, matrixStats

Suggests rstanarm, brms, testthat (>= 3.0.0), ggplot2, knitr, rmarkdown

License MIT + file LICENSE

URL <https://github.com/CoryMcCartan/conformalbayes>,
<https://corymccartan.com/conformalbayes/>

BugReports <https://github.com/CoryMcCartan/conformalbayes/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Cory McCartan [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-6251-669X>>)

Maintainer Cory McCartan <mccartan@psu.edu>

Repository CRAN

Date/Publication 2025-07-25 07:50:02 UTC

Contents

loo_conformal	2
predictive_interval.conformal	3

Index**5**

loo_conformal	<i>Enable leave-one-out conformal predictive intervals for a fit model</i>
---------------	--

Description

Prepares for jackknife(+) conformal prediction by performing Pareto-smoothed importance sampling to yield leave-one-out residuals.

Usage

```
loo_conformal(fit, ...)

## Default S3 method:
loo_conformal(
  fit,
  truth,
  chain = NULL,
  trans = function(x) x,
  inv_trans = function(x) x,
  est_fun = c("mean", "median"),
  ...
)

## S3 method for class 'stanreg'
loo_conformal(
  fit,
  trans = function(x) x,
  inv_trans = function(x) x,
  est_fun = c("mean", "median"),
  ...
)

## S3 method for class 'brmsfit'
loo_conformal(
  fit,
  trans = function(x) x,
  inv_trans = function(x) x,
  est_fun = c("mean", "median"),
  ...
)
```

Arguments

fit	Model fit; an object with posterior_predict() and log_lik() methods. Can also be an array of posterior predictions.
...	Ignored.

truth	True values to predict. Not required for rstanarm or brms models.
chain	An integer vector identifying the chain numbers for the posterior draws. Should be provided if multiple chains are used.
trans, inv_trans	A pair of functions to transform the predictions before performing conformal inference.
est_fun	Whether to use the posterior mean (the default) or median as a point estimate.

Value

A modified fit object with an additional class conformal. Calling `predictive_interval()` on this new object will yield conformal intervals.

References

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., & Gabry, J. (2015). Pareto smoothed importance sampling. [arXiv preprint arXiv:1507.02646](#).

Examples

```
# takes several seconds
if (requireNamespace("rstanarm", quietly=TRUE)) suppressWarnings({
  library(rstanarm)
  # fit a simple linear regression
  m = stan_glm(mpg ~ disp + cyl, data=mtcars,
    chains=1, iter=500,
    control=list(adapt_delta=0.999), refresh=0)

  loo_conformal(m)
})
```

predictive_interval.conformal

Jackknife(+) predictive intervals

Description

Construct finite-sample calibrated predictive intervals for Bayesian models, following the approach in Barber et al. (2021). By default, the intervals will also reflect the relative uncertainty in the Bayesian model, using the locally-weighted conformal methods of Lei et al. (2018).

Usage

```
## S3 method for class 'conformal'
predictive_interval(object, probs = 0.9, plus = NULL, local = TRUE, ...)
```

Arguments

object	A fitted model which has been passed through <code>loo_conformal()</code>
probs	The coverage probabilities to calculate intervals for. Empirically, the coverage rate of the constructed intervals will generally match these probabilities, but the theoretical guarantee for a probability of $1 - \alpha$ is only for coverage of at least $1 - 2\alpha$, and only if <code>plus=TRUE</code> (below).
plus	If TRUE, construct jackknife+ intervals, which have a theoretical guarantee. These require higher computational costs, which scale with both the number of training and prediction points. Defaults to TRUE when both of these numbers are less than 500.
local	If TRUE (the default), perform locally-weighted conformal inference. This will inflate the width of the predictive intervals by a constant amount across all predictions, preserving the relative amount of uncertainty captured by the model. If FALSE, all predictive intervals will have (nearly) the same width.
...	Further arguments to the <code>posterior_predict()</code> method for object.

Value

A matrix with the number of rows matching the number of predictions. Columns will be labeled with a percentile corresponding to `probs`; e.g. if `probs=0.9` the columns will be 5% and 95%.

References

- Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2021). Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1), 486-507.
- Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., & Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523), 1094-1111.

Examples

```
# takes several seconds
if (requireNamespace("rstanarm", quietly=TRUE)) suppressWarnings({
  library(rstanarm)
  # fit a simple linear regression
  m = stan_glm(mpg ~ disp + cyl, data=mtcars,
    chains=1, iter=500,
    control=list(adapt_delta=0.999), refresh=0)

  m = loo_conformal(m)
  # make predictive intervals
  predictive_interval(m)
})
```

Index

`loo_conformal`, [2](#)

`loo_conformal()`, [4](#)

`posterior_predict()`, [4](#)

`predictive_interval()`, [3](#)

`predictive_interval.conformal`, [3](#)