

# Introduction

Jackknife+[1] is a conformal prediction method to construct prediction intervals without distributional or algorithmic assumptions. Here, we only assume the training/testing data to be i.i.d. and the algorithm to be invariant to permutation of the training set. In this package, we assume the algorithm to be linear regression. Jackknife+ constructs the prediction interval based on the sample quantile of leave-one-out (LOO) residuals. It enjoys a theoretical coverage of  $1-2a$ .

In this package, several conformal prediction algorithms, including Jackknife+, Jackknife, Jackknife-mm, and K-fold cross-validation are implemented. The conformal prediction algorithms take the training dataset as input and produces a function mapping the test data to its prediction interval.

## Installation

```
git clone:https://github.com/Wei-weiWang/jackknifeplus.git
```

Or

```
#install.packages("devtools") if don't install, please install
devtools::install_github("Wei-weiWang/jackknifeplus", dependencies = TRUE)
```

```
## Skipping install of 'jackknifeplus' from a github remote, the SHA1 (3ededc5a) has not changed since 1
## Use `force = TRUE` to force installation
```

## Functions

We have four main R functions: `jackknifeplus_c_wrapper`, `jackknife_c_wrapper`, `jackknifeplusMM_c_wrapper` and `jackknifeplusCV_c_wrapper` which are to calculate prediction intervals by four different methods respectively: Jackknife+, Jackknife, Jackknife-mm, and K-fold cross-validation. Actually, we use C code inside to make them faster because these algorithms need to calculate loops.

### Jackknife+ interval

$$\hat{C}_{n,a}^{jackknife+} = [\hat{q}_{n,a}^{-}\{\hat{\mu}_{-i}(X_{n+1}) - R_i^{LOO}\}, \hat{q}_{n,a}^{+}\{\hat{\mu}_{-i}(X_{n+1}) + R_i^{LOO}\}]$$

where  $\hat{\mu}_{-i}$  is the trained algorithm without using the  $i$ th training data.  $X_{n+1}$  is testing data.  $R_i^{LOO} = |Y_i - \hat{\mu}_{-i}(X_i)|$   $i = 1, 2, \dots, n$ ,  $\hat{q}_{n,a}^{-}$  is the  $Flooring(a(n+1))$ -th smallest value of a vector and  $\hat{q}_{n,a}^{+}$  is the  $Ceiling((1-a)(n+1))$ -th smallest value of a vector. So we need to set size of training data,  $n$ , large enough to make sure  $\hat{q}_{n,a}^{-}$  and  $\hat{q}_{n,a}^{+}$  are bigger than 0 and less than  $n+1$ . Jackknife+ ensure probability coverage of  $1-2a$ .

### Jackknife interval

$$\hat{C}_{n,a}^{jackknife} = [\hat{q}_{n,a}^{-}\{\hat{\mu}(X_{n+1}) - R_i^{LOO}\}, \hat{q}_{n,a}^{+}\{\hat{\mu}(X_{n+1}) + R_i^{LOO}\}]$$

This is original Jackknife method, which cannot guarantee any probability w.r.t  $a$ .

### Jackknife+MM interval

$$\hat{C}_{n,a}^{jackknife+MM} = [\min_i \hat{\mu}_{-i}(X_{n+1}) - \hat{q}_{n,a}^{+}\{R_i^{LOO}\}, \max_i \hat{\mu}_{-i}(X_{n+1}) + \hat{q}_{n,a}^{+}\{R_i^{LOO}\}]$$

This is Jackknife+MM interval. Apparently, it is bigger than Jackknife+ interval. Jackknife+MM interval guarantee any probability coverage as  $1-a$ , which is also bigger than Jackknife+ interval.

## Jackknife+CV interval

$$\hat{C}_{n,K,a}^{jackknife+CV} = [\hat{q}_{n,a}^-\{\hat{\mu}_{-S_{k(i)}}(X_{n+1}) - R_i^{CV}\}, \hat{q}_{n,a}^+\{\hat{\mu}_{-S_{k(i)}}(X_{n+1}) + R_i^{CV}\}]$$

We split training dataset into  $K$  subsets equally.  $\hat{\mu}_{-S_{k(i)}}$  means the model is trained without the subset that contains the  $i$ th training sample.  $R_i^{CV} = |Y_i - \hat{\mu}_{-S_{k(i)}}(X_i)|$ .

By Jackknife+CV, we can possibly train less models. But the interval may be bigger because we use less samples. The theoretical coverage of Jackknife+CV interval is  $1 - 2a - \sqrt{2/n}$ .

## Examples

```
# Example
library(jackknifeplus)
Xtrain=matrix(rnorm(200),40,5)
Ytrain=matrix(rnorm(40),40,1)
Xtest=matrix(rnorm(5),1,5)
result_1 = jackknifeplus_c_wrapper(Xtrain, Ytrain, Xtest, 0.05)
result_2 = jackknife_c_wrapper(Xtrain, Ytrain, Xtest, 0.05)
result_3 = jackknifeplusCV_c_wrapper(Xtrain, Ytrain, Xtest, 0.05, 4)
result_4 = jackknifeplusMM_c_wrapper(Xtrain, Ytrain, Xtest, 0.05)
```

[1]Barber R F, Candes E J, Ramdas A, et al. Predictive inference with the jackknife+[J]. The Annals of Statistics, 2021, 49(1): 486-507.