Stacked Conformal Prediction

Paulo C. Marques F.

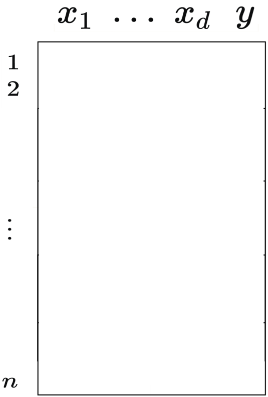
December, 2025

Insper

Our goal

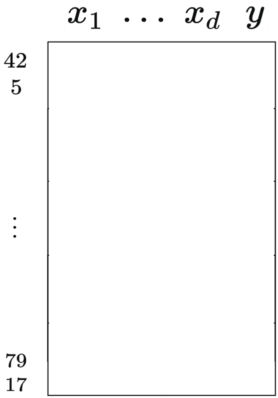
To leverage the predictive gains of model stacking with a simple meta-learner, enabling a cost-effective conformalization procedure that avoids the use of a separate calibration sample.

The method in a nutshell



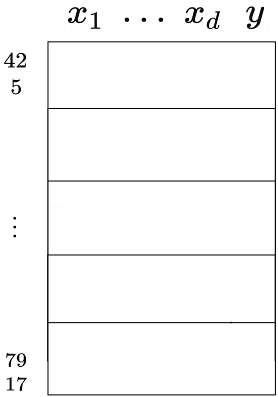
The method in a nutshell

Insper

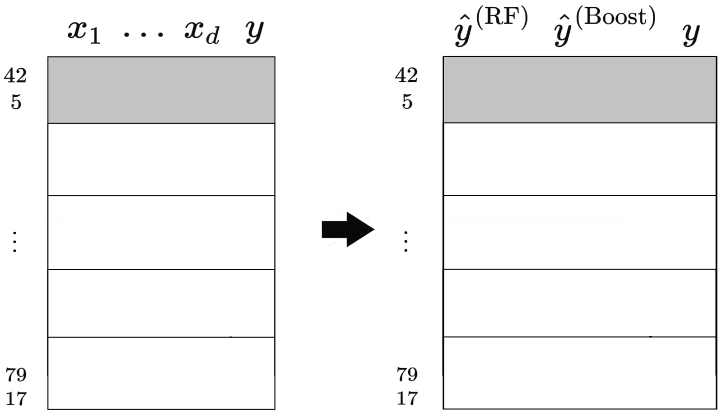


The method in a nutshell

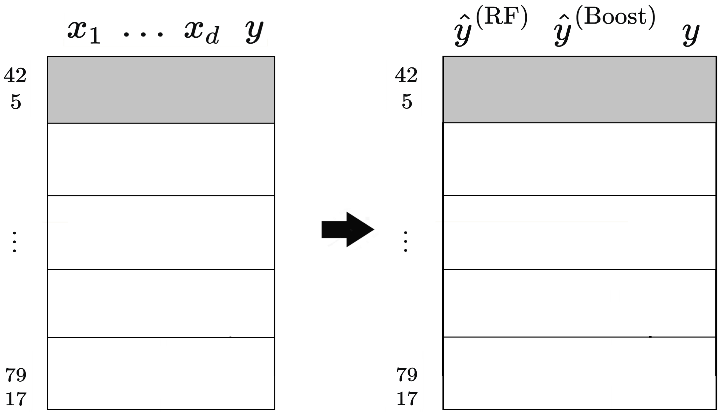
Insper



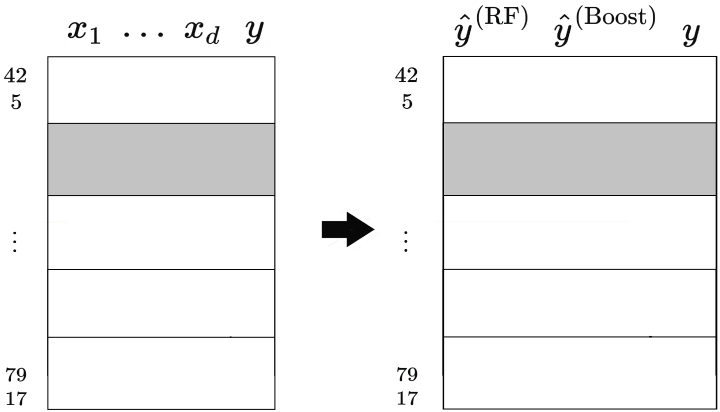
The method in a nutshell



The method in a nutshell

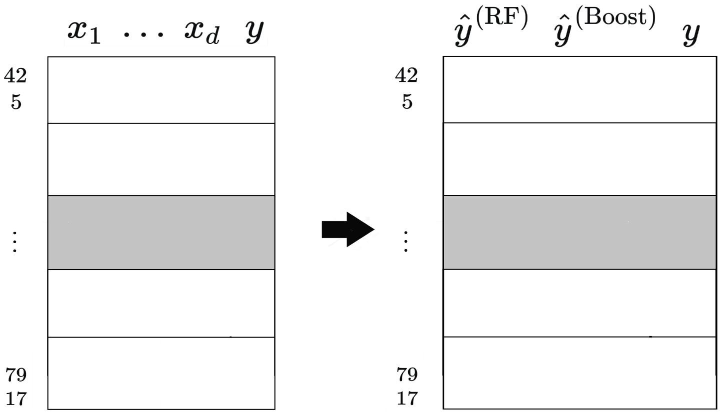


The method in a nutshell

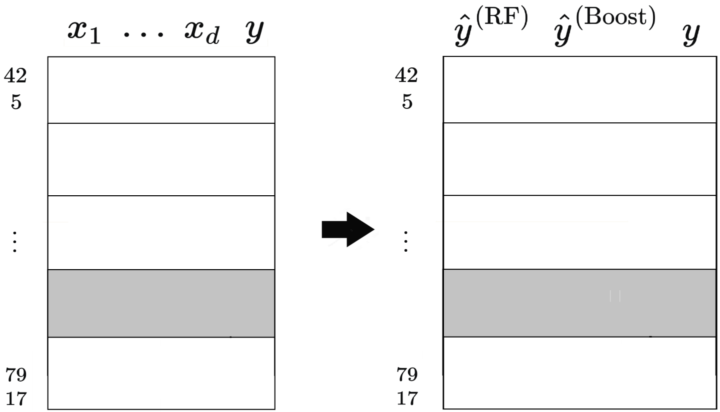


The method in a nutshell

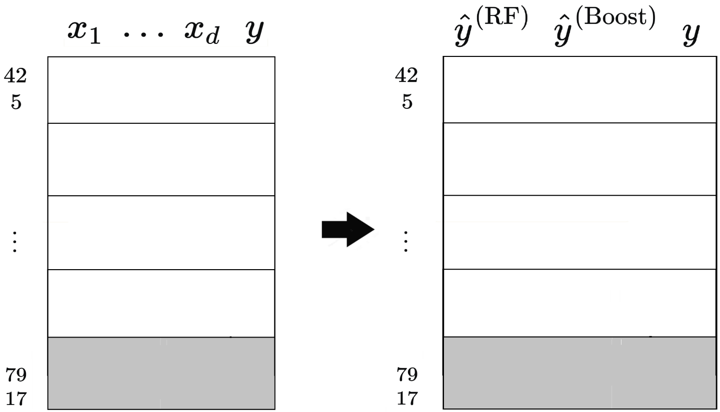
Insper



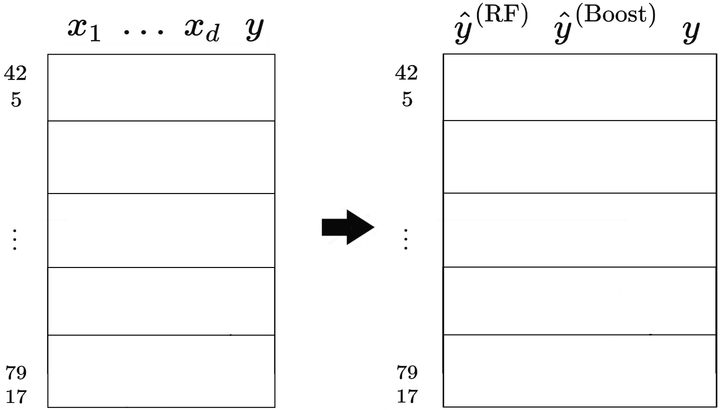
The method in a nutshell



The method in a nutshell



The method in a nutshell



Meta-learner: $y_i = \beta_0 + \beta_1 \times \hat{y}_i^{(\mathrm{RF})} + \beta_2 \times \hat{y}_i^{(\mathrm{Boost})} + \epsilon_i$

$$y = Z\beta + \epsilon \qquad \hat{\beta} = (Z^\top Z)^{-1} Z^\top y$$

The method in a nutshell

Test pair: $x_0, y_0 \in \mathbb{R}^d \times \mathbb{R}$

$$z_0 = \begin{bmatrix} 1 \\ \hat{y}_0^{(\mathrm{RF})} \\ \hat{y}_0^{(\mathrm{Boost})} \end{bmatrix} \qquad Z_+ = \begin{bmatrix} Z \\ z_0^\top \end{bmatrix} \qquad y_+ = \begin{bmatrix} y \\ y_0 \end{bmatrix}$$

The method in a nutshell

Test pair: $x_0, y_0 \in \mathbb{R}^d \times \mathbb{R}$

$$z_0 = \begin{bmatrix} 1 \\ \hat{y}_0^{(\mathrm{RF})} \\ \hat{y}_0^{(\mathrm{Boost})} \end{bmatrix} \qquad Z_+ = \begin{bmatrix} Z \\ z_0^\top \end{bmatrix} \qquad y_+ = \begin{bmatrix} y \\ y_0 \end{bmatrix}$$

$$Z_+^\top Z_+ = Z^\top Z + z_0 z_0^\top$$

The method in a nutshell

Insper

Test pair: $x_0, y_0 \in \mathbb{R}^d \times \mathbb{R}$

$$z_0 = \begin{bmatrix} 1 \\ \hat{y}_0^{(\mathrm{RF})} \\ \hat{y}_0^{(\mathrm{Boost})} \end{bmatrix} \qquad Z_+ = \begin{bmatrix} Z \\ z_0^\top \end{bmatrix} \qquad y_+ = \begin{bmatrix} y \\ y_0 \end{bmatrix}$$

$$Z_+^\top Z_+ = Z^\top Z + z_0 z_0^\top$$

$$\boxed{(G + uv^\top)^{-1} \; = \; G^{-1} - \frac{G^{-1}uv^\top G^{-1}}{1 + v^\top G^{-1}u} \qquad \text{(Sherman-Morrison (1949))}}$$

The method in a nutshell

Insper

Test pair: $x_0, y_0 \in \mathbb{R}^d \times \mathbb{R}$

$$z_0 = \begin{bmatrix} 1 \\ \hat{y}_0^{(\mathrm{RF})} \\ \hat{y}_0^{(\mathrm{Boost})} \end{bmatrix} \qquad Z_+ = \begin{bmatrix} Z \\ z_0^\top \end{bmatrix} \qquad y_+ = \begin{bmatrix} y \\ y_0 \end{bmatrix}$$

$$Z_+^\top Z_+ = Z^\top Z + z_0 z_0^\top$$

$$\boxed{(G + uv^\top)^{-1} \; = \; G^{-1} - \frac{G^{-1}uv^\top G^{-1}}{1 + v^\top G^{-1} u} \qquad \text{(Sherman-Morrison (1949))}}$$

$$A = (Z^T Z)^{-1} \qquad B = (Z_+^\top Z_+)^{-1} = A - \frac{A z_0 z_0^\top A}{1 + z_0^\top A z_0}$$

$$\hat{\beta}_+ = \hat{\beta} + (y_0 - z_0^\top \hat{\beta}) B z_0$$

The method in a nutshell

Insper

$$\hat{y} = Z\hat{\beta}_+ \qquad r_i = |y_i - \hat{y}_i| \qquad i = 1, \ldots, n$$

The method in a nutshell

$$\hat{y} = Z\hat{\beta}_+ \qquad r_i = |y_i - \hat{y}_i| \qquad i = 1, \ldots, n$$

$$r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)} \qquad \hat{r} = r_{(\lceil (1-\alpha)(n+1) \rceil)}$$

The method in a nutshell

Insper

$$\hat{y} = Z\hat{\beta}_+ \qquad r_i = |y_i - \hat{y}_i| \qquad i = 1, \ldots, n$$

$$r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)} \qquad \hat{r} = r_{(\lceil (1-\alpha)(n+1) \rceil)}$$

$$\hat{y}_0 = z_0^\top \hat{\beta}_+ \qquad r_0 = |y_0 - \hat{y}_0|$$

$$y_0 \in \{\text{Conformal Prediction Set}\} \quad \Leftrightarrow \quad r_0 \leq \hat{r}$$

The method in a nutshell

$$\hat{y} = Z\hat{\beta}_+ \qquad r_i = |y_i - \hat{y}_i| \qquad i = 1, \ldots, n$$

$$r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)} \qquad \hat{r} = r_{(\lceil (1-\alpha)(n+1) \rceil)}$$

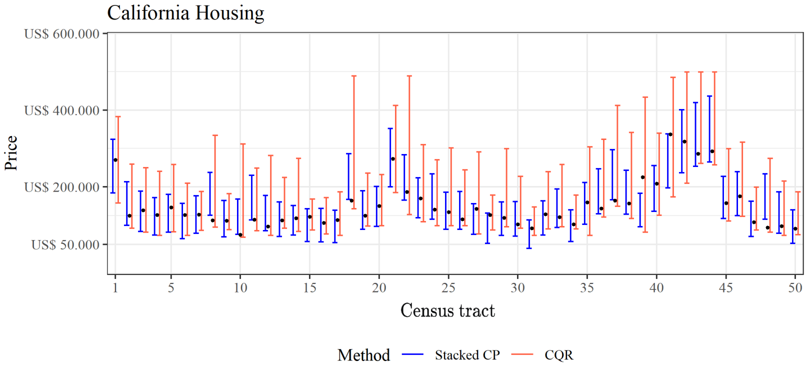
$$\hat{y}_0 = z_0^\top \hat{\beta}_+ \qquad r_0 = |y_0 - \hat{y}_0|$$

$$y_0 \in \{\text{Conformal Prediction Set}\} \quad \Leftrightarrow \quad r_0 \leq \hat{r}$$

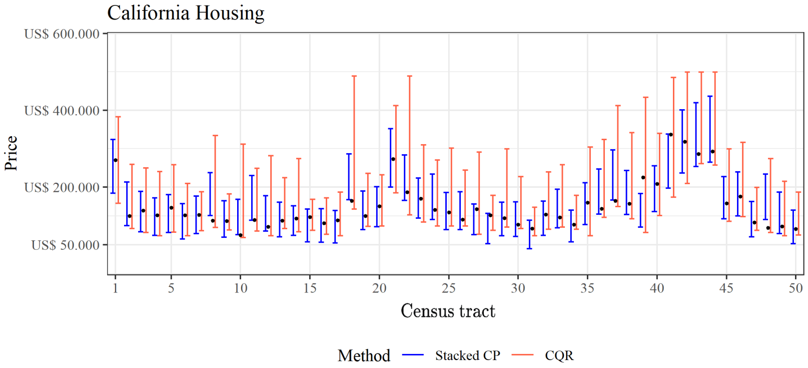
Note: Actually, we scale our conformity scores. Check the full algo in the paper.

Does it work?

Does it work?


California Housing
US$ 600.000
US$ 400.000
US$ 200.000
US$ 50.000
Price
1
5
10
15
20
25
30
35
40
45
50
Census tract
Method
Stacked CP
CQR

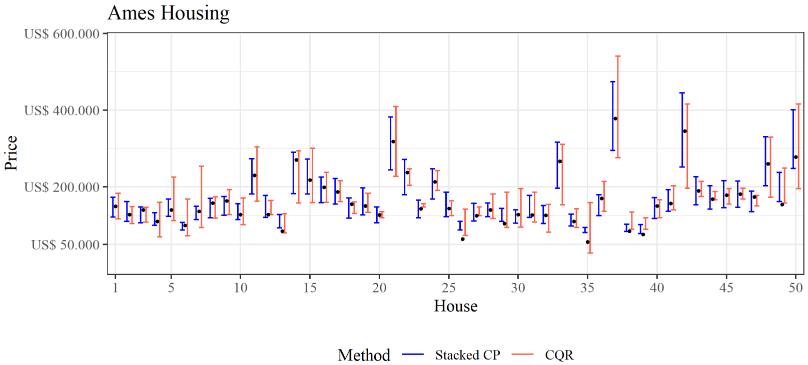
Does it work?



Nominal coverage level: 90%

Empirical coverage: 89.9%

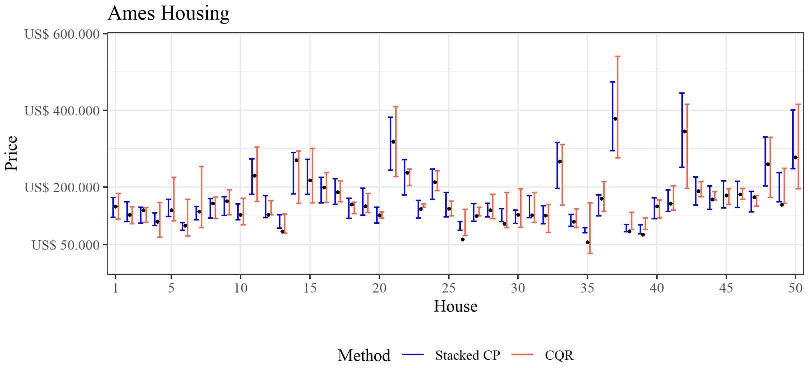
Does it work?



Ames Housing
US$ 600.000
US$ 400.000
US$ 200.000
US$ 50.000
Price
1
5
10
15
20
25
30
35
40
45
50
House
Method
Stacked CP
CQR

Does it work?

Insper



Nominal coverage level: 90%

Empirical coverage: 91.1%

But. . .

Insper


WHY DOES
IT WORK?
DATA
ALGORITHMS
OPTIMZATION
DATA
PREDICTION

Key idea

Construct an idealized totally symmetric stack which includes the future observable pair (X_{n+1}, Y_{n+1}).

Prove that for this symmetric stack that the assumed exchangeability of the data sequence is transferred to the second stack level.

Symmetric stack

Insper

Data sequence: $(X_1, Y_1), (X_2, Y_2), \ldots$

Regression setting: $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$

Future observable pair: (X_{n+1}, Y_{n+1})

$$T = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} & Y_1 \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} & Y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} & Y_n \\ X_{n+1,1} & X_{n+1,2} & \cdots & X_{n+1,d} & Y_{n+1} \end{pmatrix}$$

$\mathbb{S}_n$ denotes the set of $n \times n$ permutation matrices.

Exchangeability assumption: $T \sim \Pi T$, for every permutation matrix $\Pi \in \mathbb{S}_{n+1}$.

Insper

Divide the training sample into $K \geq 2$ folds of size $t = (n+1)/K$, assuming that $n+1$ is divisible by K, by means of an $(n+1) \times (n+1)$ random permutation matrix Q (the folding scheme matrix).

Suppose Q is uniformly distributed: $\Pr(Q = q) = 1/(n+1)!$, for every $q \in \mathbb{S}_{n+1}$.

Suppose Q and T are independent.

Block notation:

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_K \end{pmatrix},$$

in which the realizations of the matrix blocks Q_k are in $\mathbb{R}^{t \times (n+1)}$, for $k = 1, \ldots, K$.

Symmetric stack

Let ϕ be a fold indicator function, defined by

$$\phi(i) = \sum_{k=1}^{K} k \cdot I(\sum_{\ell=1}^{t} (Q_k)_{\ell,i} = 1),$$

meaning that $\phi(i)$ determines the fold number k to which the i-th training sample unit has been assigned, for $i = 1, \ldots, n+1$.

Define, for $k = 1, \ldots, K$, the k-th fold exclusion matrix

$$Q_{\setminus k} = \begin{pmatrix} Q_1 \\ \vdots \\ Q_{k-1} \\ Q_{k+1} \\ \vdots \\ Q_K \end{pmatrix},$$

whose realizations are in $\mathbb{R}^{(n+1-t)\times(n+1)}$.

Symmetric stack

The stack is built from $M \geq 1$ base learning methods.

For $m = 1, \ldots, M$, we have prediction functions

$$\hat{\mu}_m : \mathbb{R}^{(n+1-t)\times(d+1)} \times \mathbb{R}^d \to \mathbb{R},$$

and we assume that each learning method treats its training data $S \in \mathbb{R}^{(n+1-t)\times(d+1)}$ symmetrically, so that

$$\hat{\mu}_m(S, x) = \hat{\mu}_m(\Pi S, x),$$

for every permutation matrix $\Pi \in \mathbb{S}_{n+1-t}$ and each $x \in \mathbb{R}^d$.

The stack base-learners make predictions

$$Z_i = (Z_{i,1}, \ldots, Z_{i,M}) \in \mathbb{R}^M,$$

in which $Z_{i,m} = \hat{\mu}_m\big(Q_{\setminus \phi(i)} T, X_i\big)$, for $i = 1, \ldots, n+1$.

Symmetric stack

When the learning method involves some form of randomization - such as the bootstrap process in Random Forests, or the stochastic gradient descent optimization in Deep Neural Networks - we assume, without loss of generality, that the seed of the underlying pseudo-random number generator is set using a symmetric hash function of the training data S.

Symmetric stack

Insper

Proposition

The second-level random pairs $(Z_1, Y_1), \ldots, (Z_{n+1}, Y_{n+1})$ *are exchangeable for a symmetric stack.*

Insper

Proposition

Let $\hat{\psi}_n^{(y)}$ be a meta-learner trained from

$$\{(Z_1, Y_1), \ldots, (Z_n, Y_n), (Z_{n+1}, y)\},$$

for $y \in \mathbb{R}$, and suppose the order of the $n+1$ pairs in this sample is irrelevant for the construction of $\hat{\psi}_n^{(y)}$. For a conformity function $\rho : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, define the conformity scores $R_i^{(y)} = \rho(Y_i, \hat{\psi}_n^{(y)}(Z_i))$, for $i = 1, \ldots, n+1$, and let $R_{(1)}^{(y)} \leq R_{(2)}^{(y)} \leq \cdots \leq R_{(n)}^{(y)}$ denote the ordered conformity scores among $\{R_1^{(y)}, R_2^{(y)} \ldots, R_n^{(y)}\}$. Choosing a nominal miscoverage level $0 < \alpha < 1$ such that $1 \leq \lceil (1-\alpha)(n+1) \rceil \leq n$, we have that

$$\Pr\Big(Y_{n+1} \in C_{n+1}^{(\alpha)}\Big) \geq 1 - \alpha,$$

in which we defined the random prediction set

$$C_{n+1}^{(\alpha)} = \Big\{ y \in \mathbb{R} : R_{n+1}^{(y)} \leq R_{(\lceil (1-\alpha)(n+1) \rceil)}^{(y)} \Big\}.$$

Feasible stack

The symmetric stack is an oracle construct that cannot be implemented in practice, since at training time we do not know the observed value of the future response Y_{n+1}.

A *feasible stack* is attained by removing the future observable pair (X_{n+1}, Y_{n+1}) from the training sample.

In doing so, we break the distributional symmetry of the stack, but if the predictions made by the stack base-learners stay stable after this single sample unit removal, we can argue that a marginal validity property still holds approximately.

Feasible stack

Insper

Suppose that in this feasible stack the first level prediction Z_{n+1} is now made by base-learners trained on the whole training sample and let $\tilde{R}_{n+1}^{(Y_{n+1})}$ and $\tilde{R}_{(\lceil(1-\alpha)(n+1)\rceil)}^{(Y_{n+1})}$ be the corresponding random conformity scores pertaining to the feasible stack.

Proposition

Given $\epsilon > 0$, if there is a $\delta = \delta(\epsilon) > 0$ such that

$$\Pr\left(\max\left\{\left|\tilde{R}_{n+1}^{(Y_{n+1})} - R_{n+1}^{(Y_{n+1})}\right|, \left|\tilde{R}_{(\lceil(1-\alpha)(n+1)\rceil)}^{(Y_{n+1})} - R_{(\lceil(1-\alpha)(n+1)\rceil)}^{(Y_{n+1})}\right|\right\} < \epsilon/2\right) \geq 1-\delta,$$

then

$$\Pr\left(Y_{n+1} \in \tilde{C}_{n+1}^{(\alpha)}\right) \geq 1 - \alpha - \delta - h(\epsilon),$$

in which

$$\tilde{C}_{n+1}^{(\alpha)} = \left\{y \in \mathbb{R} : \tilde{R}_{n+1}^{(y)} \leq \tilde{R}_{(\lceil(1-\alpha)(n+1)\rceil)}^{(y)}\right\}$$

and

$$h(\epsilon) = \Pr\left(R_{(\lceil(1-\alpha)(n+1)\rceil)}^{(Y_{n+1})} - \epsilon < R_{n+1}^{(Y_{n+1})} \leq R_{(\lceil(1-\alpha)(n+1)\rceil)}^{(Y_{n+1})}\right).$$

Published paper and software repo

Insper

Proceedings of Machine Learning Research 266:1–12, 2025 Conformal and Probabilistic Prediction with Applications

Stacked conformal prediction

Paulo C. Marques F. PAULOCMF1@INSPER.EDU.BR
Insper Institute of Education and Research, Rua Quatá 300, São Paulo 04546-042, Brazil

http://github.com/paulocmarquesf/stacked_cp

Thank you very much!