Flexible and Systematic Uncertainty Estimation with Conformal Prediction via the MAPIE library

Thibault Cordier[1] TCORDIER@QUANTMETRY.COM
Vincent Blot[1,3] VBLOT@QUANTMETRY.COM
Louis Lacombe[1] LLACOMBE@QUANTMETRY.COM
Thomas Morzadec[1] TMORZADEC@QUANTMETRY.COM
Arnaud Capitaine ARNAUD.GC.CAPITAINE@GMAIL.COM
Nicolas Brunel[1,2] NBRUNEL@QUANTMETRY.COM
[1] *Quantmetry, 52, rue d'Anjou, 75008, Paris, France*
[2] *Laboratoire de Mathématiques et de Modélisation d'Evry, ENSIIE, Paris-Saclay University*
[3] *Laboratoire interdisciplinaire des sciences du numérique, CNRS, Paris-Saclay University*

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

Conformal prediction (CP) is an attractive theoretical framework for estimating the uncertainties of any predictive algorithms as its methodology is general and systematic with few assumptions. CP methods can be abstracted into building blocks that can be deployed on any type of data, model, or task. In this work, we contribute to the wide diffusion of the CP framework by developing the library MAPIE[1] that implements such principles and can address seamlessly different tasks (e.g. classification, regression, time-series) and in different settings (split and cross-conformal). All these concepts are under a common umbrella with an emphasis on readability, transparency, and reliability, hence supporting the principles of trustworthy AI. An original feature of MAPIE is to offer the possibility of designing tailored-made non-conformity scores in particular p-normalized residual non-conformal scores that can be defined to account for asymmetric errors. We show theoretically the marginal coverage guarantee in several settings. We highlight through applications the interest of choosing different non-conformity scores for tabular data when considering local coverage.

Keywords: conformal prediction, uncertainty quantification, open-source library, non-conformity score, homoscedasticity, heteroscedasticity

1. Introduction

Quantifying the uncertainties of ML model predictions is of crucial importance for developing and deploying reliable artificial intelligence (AI) systems. Uncertainty quantification (UQ) involves all the stakeholders who develop and use AI models. First, UQ allows the designers of AI systems to better understand the predictive power of their model and assess the validity of the model predictions on new data points. Second, UQ allows AI operators, such as business stakeholders, to optimize the risk management when making business decisions based on AI system predictions. Third, UQ helps AI regulators to assess the compliance of the AI system with the regulation in force. Fourth, UQ allows the AI systems to be more transparent and trustworthy for people impacted by the decisions made from AI.

[1] https://github.com/scikit-learn-contrib/MAPIE/

There is therefore a strong need for libraries of uncertainty quantification that respect three fundamental pillars. First, implemented methods have to be model and use case agnostic in order to address all relevant use cases tackled in industry, such as natural language processing, time series, or computer vision, using state-of-the-art ML models, like neural networks or gradient boosting models. Second, methods must have strong theoretical guarantees at least on the marginal coverage (and possibly on the conditional coverage) of the estimated uncertainties with as little assumption on the data or the model as possible. This ensures AI system designers, operators, and regulators to be confident about the predictions provided by the models. Third, libraries need to be open-source and respect state-of-the-art programming standards to develop trustworthy AI systems.

Resampling or undersampling methods have been used for a few decades to estimate the robustness of predictions (Quenouille, 1956; Efron, 1979). Among other, bootstrap is probably the most commonly used methods since it is easy to implement and allow the user to easily obtain confidence intervals associated with the predictions. However, the standard jackknife technique suffers from instabilities in some cases and can be unusable on practical use cases when the number of training data samples is high.

Although modern resampling techniques have been implemented in recent R packages (Tibshirani, 2022), they are only recently implemented in Python and remain outside the standard scikit-learn framework (Pedregosa et al., 2011). Some scikit-learn regressors allow the users to estimate confidence intervals associated with the model predictions but they are either simple regressors, such as the Bayesian Ridge model, or based on quantile regression for gradient boosting. IBM's UQ360 (Ghosh et al., 2021) and nonconformist (Linusson, 2022) libraries are noteworthy as they aim at incorporating several complementary UQ methods. More recently, Fortuna library (Detommaso et al., 2023) provides calibration and conformal methods following the model of the MAPIE library.

Since 2021, we are developing the MAPIE (Model Agnostic Prediction Interval Estimator) library in order to address the three aforementioned pillars (Taquet et al., 2022)[2]. MAPIE is an open-source Python library hosted on scikit-learn-contrib. It follows the scikit-learn guidelines; the only technical requirement is to have a scikit-learn API and accepts base scikit-learn-compatible estimators. Importantly, MAPIE implements conformal prediction methods for regression and classification settings and is therefore model and use case agnostic. Conformal prediction methods allow MAPIE to have mathematical guarantees on the marginal coverages on the prediction intervals.

In this paper, we provide an overview of the MAPIE library, the theoretical framework, and the practical implementation of conformal prediction methods through a wide variety of applications ranging from regression to classification. In particular, we show theoretical guarantees for cross-conformal methods based on more general non-conformity scores build upon the work of Foygel Barber et al. (2021). Section 2 presents a overview of conformal prediction methods including those implemented in MAPIE. Section 3 is devoted to extending the marginal coverage to a larger number of non-conformity scores. Section 4 presents those that have been implemented in MAPIE. Section 5 describes MAPIE in practice by listing the main input parameters and how to use them. Section 6 presents illustrative examples. Section 7 concludes with our perspectives.

[2] The current work is an extension of the DFUQ workshop paper (ICML) with significant and original new materials.

2. A general framework of conformal predictions

One of the important contributions of MAPIE is to implement the state-of-the-art conformal prediction methods for regression and classification settings which makes it model agnostic. Before describing these methods, we briefly present the mathematical framework of conformal prediction.

Let assume that you have a training dataset $(X, Y) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ under an unknown distribution $P_{X,Y}$ with data-exchangeability assumption. We assume that $Y = \mu(X) + \epsilon$ where μ is the predictor that is classically estimated by ML models, and $\epsilon_i \sim P_{Y|X}$ is the noise. For any risk level $\alpha \in (0; 1)$, we aim at constructing a prediction interval or a prediction set $\hat{C}_{n,\alpha}(X_{n+1})$ for a new observation (X_{n+1}, Y_{n+1}) such that:

$$1 - \alpha \leq P\{Y_{n+1} \in \hat{C}_{n,\alpha}(X_{n+1})\} \leq 1 - \alpha + \frac{1}{n+1} \tag{1}$$

In words, for a typical risk level α of 10%, we want to construct prediction sets that contain the true target value for almost exactly 90% of the new test data points under the assumption of exchangeability. This property is known as *marginal coverage.*

To achieve this objective, MAPIE proposes two different approaches depending on whether one wants to calibrate an existing model or to use resampling methods to estimate the prediction set. We will present them starting with the *split-conformal prediction* method, then focusing on the *cross-conformal prediction* methods. Finally, we consider the case where the exchangeability assumption is not respected.

2.1. Split-conformal prediction methods

If one wants to calibrate an existing model, the *split-conformal prediction* methods, also known as *inductive conformal prediction*, is the best approach to achieve marginal coverage (Papadopoulos et al., 2002; Lei et al., 2018). The construction of the conformal predictions can be divided into two steps, first the training of the model, then the computation of the non-conformity scores on the calibration data to compute the predictions sets. The methodology is described in Algorithm 1.

Split-conformal prediction is an attractive technique due to its computational efficiency, requiring the model to be fitted only once. This last point is very important when it comes to calibrating very large artificial neural networks for which the learning cost is not negligible. One would prefer to use a pre-trained model and apply a split-conformal prediction methodology. However, it incurs a statistical efficiency trade-off since it necessitates splitting the data into training and calibration datasets. Another way to avoid giving up on statistical efficiency would be to consider *cross-conformal prediction* which avoids data splitting by using resampling methods.

2.2. Cross-conformal prediction methods

Cross conformal prediction methods are relevant solutions that use a reasonable number of model fits, while using all data for both model fitting and calibration thanks to resampling methods. They therefore benefit from a compromise between statistical and computational efficiency. The general methodology is similar to that of the split-conformal prediction, but with some subtleties as described in Algorithm 2.

Algorithm 1: Split Conformal Prediction Method

- Let assume you have a model $\hat{\mu}$ trained on $\mathcal{D}_m = \{(X_1, Y_1), ..., (X_m, Y_m)\}$ and a calibration dataset $\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ composed of data which have not been used during the training.
- Choose a non-conformity score function $s(x, y)$ that quantifies how well an observation x conforms with a target y. The higher the value, the more atypical the point.
- Compute the non-conformity scores on the calibration dataset $\mathcal{D}_n$.
- Estimate $\hat{q}_{n,\alpha}$ the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of the empirical non-conformity scores distribution $\{s_1 = s(X_1, Y_1), \ldots, s_n = s(X_n, Y_n)\}$ associated with the risk level α.
- Construct prediction intervals or prediction sets $\hat{C}_{n,\alpha}(X_{n+1})$ for new test points X_{n+1} based on this quantile as follows: $\hat{C}_{n,\alpha}(X_{n+1}) = \{y : s(X_{n+1}, y) \leq \hat{q}_{n,\alpha}\}$

MAPIE has three state-of-the-art conformal prediction methods such as Jackknife+, CV+ and Jackknife+-after-bootstrap. Here, we propose to briefly examine these different state-of-the-art conformal prediction methods (we refer the reader to consult the Appendix A for more theoretical details).

Jackknife+ Presented by Foygel Barber et al. (2021), the Jackknife+ method is based on the standard Jackknife, which constructs a set of *leave-one-out* models. Unlike the standard Jackknife method which returns a prediction interval centered around the prediction of the model trained on the entire dataset, the so-called Jackknife+ method also uses the predictions from all leave-one-out models on the new test point to take the variability of the predictive function into account.

CV+ In order to reduce the computational cost, one can adopt a *cross-validation* approach instead of a leave-one-out approach, with the so-called CV+ method. As pointed out by Foygel Barber et al. (2021), Jackknife+ can be considered as a special case of CV+ with $K = n$. In practice, this method results in slightly wider prediction intervals and is therefore likely more conservative, but gives a reasonable compromise for large datasets when the Jackknife+ method is unfeasible.

Algorithm 2: Cross Conformal Prediction Method

- Let assume you have a dataset $\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ for both the training and calibration steps.
- With respect to the chosen resampling method, fit the predictive models on the training subset and compute the non-conformity scores on the calibration subset.

Jackknife+-after-bootstrap An alternative way to reduce the computational cost is to adopt a *bootstrap* approach instead of cross-validation, called the Jackknife+-after-bootstrap method, offered by Kim et al. (2020). The latter can be considered as an alternative case of CV+ where the resampling step is performed with replacement. It seems that this method results in wider prediction intervals (the uncertainty is higher than CV+) because the models' prediction spread is higher.

2.3. Conformal methods without data-exchangeability assumption

The methods implemented in MAPIE assume that data are exchangeable. This hypothesis is hardly reasonable for dynamical time series, thus requiring a specific method. The method implemented in MAPIE is based on the algorithm *Ensemble Batch Prediction Intervals* (EnbPI) (Xu and Xie, 2020). In this method, the assumption on data exchangeability is replaced with assumptions on the residuals of the estimators (errors process and estimations quality). Moreover, the coverage guaranty is not absolutely but approximately valid up to these assumptions validity. The corresponding algorithm is very closed to Jackknife+-after-bootstrap. It is notably based on bootstrapping (with a block bootstrap that suits to time series). Furthermore, the non-conformity scores are updated during the prediction process. So they can dynamically take into account, for example, an increase in the variance of the residuals or a model deterioration.

3. Extension of the marginal coverage guarantee to a larger number of non-conformity score functions

The original contribution of this paper that we have implemented in MAPIE is an extension of the marginal coverage guarantee to a larger number of non-conformity scores. Indeed, for the regression task, the computation of the non-conformity score was limited to the residuals. We extend this property to a larger number of non-conformity scores by introducing the concepts of perturbed prediction function and signed loss score function.

3.1. Marginal coverage with signed loss score functions

We consider as a *signed loss score function* $f(\hat{y}, y)$, where $\hat{y} = \hat{\mu}(x)$, any function that evaluate the positive non-conformity score (when $\hat{y}$ is above y) or the negative non-conformity score (when $\hat{y}$ is below y). We assume that it is monotonically increasing on $\hat{y}$ and monotonically decreasing on y. In words, a target y is so considered atypical for a reference $\hat{y}$ if its absolute score value is high. The *perturbed prediction function* $g(\hat{s}, \hat{y})$ is the reciprocal function that reconstructs the target y given the reference $\hat{y}$ and the signed loss score $\hat{s}$ such that:

$$y = g(\hat{s}, \hat{y}) = g(f(\hat{y}, y), \hat{y}) \quad \text{or} \quad \hat{s} = f(\hat{y}, y) = f(\hat{y}, g(\hat{s}, \hat{y})) \tag{2}$$

This formalisation allows us to define the prediction intervals $\hat{C}_{n,\alpha}(X_{n+1})$ for new test points X_{n+1} for which we have predicted the target $\hat{\mu}(X_{n+1})$ as follows:

$$\begin{aligned} \hat{C}_{n,\alpha}(X_{n+1}) &= \{y : \hat{q}^-_{n,\alpha} \leq f(\hat{\mu}(X_{n+1}), y) \leq \hat{q}^+_{n,\alpha}\} \\ &= \{y : g(\hat{q}^-_{n,\alpha}, \hat{\mu}(X_{n+1})) \leq y \leq g(\hat{q}^+_{n,\alpha}, \hat{\mu}(X_{n+1})\} \\ &= \{y : \hat{y}^-_{n,\alpha}(X_{n+1}) \leq y \leq \hat{y}^+_{n,\alpha}(X_{n+1})\} \end{aligned} \tag{3}$$

where $\hat{y}^+_{n,\alpha}(X_{n+1})$ and $\hat{y}^-_{n,\alpha}(X_{n+1})$ are respectively the lower and upper bound of the prediction interval with a risk level α. They can be reconstructed with the non-conformity score quantiles $\hat{q}^+_{n,\alpha}$ and $\hat{q}^-_{n,\alpha}$ according to the perturbed prediction function g and $\hat{\mu}(X_{n+1})$.

We show that, for any conformal prediction method that satisfies marginal coverage with residual non-conformity scores, this property holds for any signed loss score function that can be reconstructed with a perturbed prediction function.

Theorem 1 (Global marginal coverage guarantee) *We state that, for any signed loss score function $f(\hat{y}, y)$ monotonically increasing on $\hat{y}$ and monotonically decreasing on y (the higher the absolute value, the more atypical the point), for any conformal prediction methods in Table 3.1, the prediction interval satisfies the marginal coverage:*

$$P\{Y_{n+1} \in \hat{C}_{n,\alpha}(X_{n+1})\} \gtrsim 1 - \alpha$$

We provide a comprehensive proof of several conformal prediction methods in Appendix C and summarize the main claims in Table 3.1.

Method	**Theoretical coverage**	**Training cost**	**Evaluation cost**
Naïve	No guarantee	1	n_{test}
Split	$\geq 1-\alpha$	1	n_{test}
Jackknife	No guarantee	n	n_{test}
Jackknife+	$\geq 1-2\alpha$	n	$n \times n_{\text{test}}$
Jackknife-minmax	$\geq 1-\alpha$	n	$n \times n_{\text{test}}$
CV	No guarantee	K	n_{test}
CV+	$\geq 1-2\alpha$	K	$K \times n_{\text{test}}$
CV-minmax	$\geq 1-\alpha$	K	$K \times n_{\text{test}}$
Jackknife-aB+	$\geq 1-2\alpha$	K	$K \times n_{\text{test}}$
Jackknife-aB-minmax	$\geq 1-\alpha$	K	$K \times n_{\text{test}}$

Table 1: Theoretical marginal coverage and reminder of the training cost and the evaluation cost for conformal prediction methods (Foygel Barber et al., 2021). These properties remain valid for any signed loss score, as shown in Theorem 1. α is the risk level, n and n_{test} the number of train and test data and K the number of fold used in cross-validation and bootstrap.

4. Implementation of state-of-the-art CP methods in MAPIE

4.1. Non-conformity scores for regression

Based on the previous statement, we have so far provided two non-conformity score functions for the regression in MAPIE: the standard *absolute residual* non-conformity score and the *p-normalized residual* non-conformity score.

Absolute residual non-conformity score As commonly used, the absolute residual non-conformity score function is the standard metric for estimating the non-conformity between a target y and a prediction $\hat{\mu}(x)$ based on their difference:

$$s(x,y) = |y - \hat{\mu}(x)| \implies \begin{cases} f(\hat{\mu}(x), y) & = & y - \hat{\mu}(x) = \hat{s} \\ g(\hat{s}, \hat{\mu}(x)) & = & \hat{\mu}(x) + \hat{s} \end{cases} \tag{4}$$

Normalized residual non-conformity score of degree p In order to take into account the order of magnitude of the prediction in the uncertainty estimation, the p-normalized residual non-conformity score function has been proposed based on scaling the absolute residual nonconformity score to the prediction:

$$\forall p \in \mathbb{R}, \quad s_p(x,y) = \left|\frac{y - \hat{\mu}(x)}{\hat{\mu}(x)^{p/2}}\right| \implies \begin{cases} f(\hat{\mu}(x), y) & = & \frac{y-\hat{\mu}(x)}{\hat{\mu}(x)^{p/2}} = \hat{s} \\ g(\hat{s}, \hat{\mu}(x)) & = & \hat{\mu}(x) + \hat{s}\hat{\mu}(x)^{p/2} \end{cases} \tag{5}$$

This process generates prediction intervals that are wider or narrower depending on the order of magnitude of the prediction and thus provides an adaptive conformal prediction in the target space as follows:

$$\hat{C}_{n,\alpha}(X_{n+1}) = \{y : \hat{\mu}(X_{n+1}) + \hat{q}^{-}_{n,\alpha}\hat{\mu}(X_{n+1})^{p/2} \leq y \leq \hat{\mu}(X_{n+1}) + \hat{q}^{+}_{n,\alpha}\hat{\mu}(X_{n+1})^{p/2}\} \tag{6}$$

For the following, we can notice that for $p = 0$, the p-normalized residual non-conformity score is equivalent to the absolute score. Moreover, for $p = 1$, the residual is normalized by $\sqrt{\hat{\mu}(x)}$ and by $\hat{\mu}(x)$ for $p = 2$.

4.2. When heteroscedasticity occurs in regression

In MAPIE, one of the objectives is to manage the heteroscedasticity of the data in order to estimate relevant prediction intervals for each observation:

- When using the standard absolute residual non-conformity score, the width of prediction intervals does not vary locally since, for any observation x, it depends only on a constant quantile $\hat{q}_{n,\alpha}(x) = \hat{q}_{n,\alpha}$ and fails to undertake heteroscedastic noise.
- Assuming that the heteroscedastic noise depends on the prediction, choosing a p-normalized residual non-conformity score allows us to vary the width of the prediction intervals such that $\hat{q}^{\pm}_{n,\alpha}(x) = q^{\pm}_{n,\alpha} \times \hat{\mu}(x)^{p/2}$, thus allowing us to adapt them to the order of magnitude of the prediction.
- In the case where the heteroscedastic noise depends on the observation, we provide more methods for estimating better interval widths as the Conformalized Quantile Regression (Romano et al., 2019). It uses quantile regressors $\hat{q}_{n,\alpha}(x)$ with different quantile values to estimate the prediction bounds and the residuals of these methods is used to create the guaranteed coverage value.

It is important to specify that the choice of the non-conformity score depends on the nature of the noise and thus on the use case. An upstream study allows to know which method is the most relevant to manage the heteroscedasticity of the data. In particular for the p-normalized residual non-conformity score, the choice of the degree p can be motivated by an analysis of the distribution of the residuals or by a preference to better cover the upper or lower target values. We will illustrate this point in the Section 5.

4.3. Conformalized Quantile Regression (CQR)

This last method has newly been implemented in MAPIE following the published paper by Romano et al. (2019). This method aims to tackle the issue of conformal predictions with heteroscedastic data. A homoscedastic dataset can be described by variance that is uniform, therefore leading to prediction intervals of the same size, and this is one of the main drawbacks of the previously mentioned methods when using absolute residual non-conformity scores.

CQR is a method that makes use of quantile regression techniques (it is important to note that it is outside the scope of the Algorithm 1 and it has a dedicated Algorithm 3). It requires fitting three quantile regressions at different quantile levels; for a coverage of $(1-\alpha)$, we need to fit at: $[\alpha/2, 1-(\alpha/2), 0.5]$. In a non-conformal setting, the intuition would be that of [*lower bound*, *upper bound*, *point prediction*], thereby already creating confidence intervals for the predictions. Whereas in the previous methods of conformal methods we would create a non-conformity scores using the absolute value of residuals from the point prediction, for CQR, the non-conformity scores are defined by the maximum value between the difference of the lower and upper bound fitted quantiles with the calibration point.

Algorithm 3: Conformalized Quantile Regression

```
Require: $\mu$: quantile regression model , $X$: independent variable, $y$: dependent variable
    and $\alpha$: such that the target coverage is $(1-\alpha)$
Ensure: $\hat{\mu}(X_{n+1})$: fitted model with point prediction, $\hat{q}^{+}_{n,\alpha}\{R_i\}$: the interval for the prediction.
1: Split $X$ into a training, calibration and test dataset.
2: Fit $\mu$ on the training set at the following three quantile levels: $[\alpha/2, 1-(\alpha/2), 0.5]$ which
   we will respectively call $[\hat{\mu}_{low}, \hat{\mu}_{high}, \hat{\mu}_{pred}]$
3: Make predictions for $\hat{\mu}_{low}, \hat{\mu}_{high}$ on the calibration set and calculate $R_i = \max\{\hat{\mu}_{low} - Y_i, Y_i - \hat{\mu}_{high}\}$.
   Note that $R_i$ has the length of the calibration set, $n$.
4: Return $\hat{C}^{\text{cqr}}_{n,\alpha}(X_{n+1}) = [\hat{\mu}_{low}(X_{n+1}) - \hat{q}_{n,\alpha}\{R_i\}, \hat{\mu}_{high}(X_{n+1}) + \hat{q}_{n,\alpha}\{R_i\}]$
```

Alternatives to this method have been explored, by changing the way to calculate the R_i. As shown in the paper by Sesia and Candès (2020), these methods do not improve the original CQR method and they even suggest that it is better to optimize the training of the algorithm rather than spending time trying to improve the conformal method. That being said, Sousa et al. (2022) suggests a slightly different methodology to allow for more adaptiveness of the CQR by clustering the X and adding a different quantile value depending on which cluster it belongs to. Note that, as shown by Romano et al. (2019), the CQR statistically guarantees $P(Y_{n+1} \in \hat{C}_n(X_{n+1})) \geq 1-\alpha$.

4.4. Non-conformity scores for classification

To complete this overview of the methods implemented in MAPIE, four methods for multi-class classification UQ have been proposed so far: LABEL (Sadinlem et al., 2019), Adaptive Prediction Sets (Romano et al., 2020), Top-K (Angelopoulos et al., 2020) and RAPS (Angelopoulos et al., 2020). The difference between these methods is the way the non-conformity scores are computed (we refer the reader to consult the Appendix B for more theoretical details).

LABEL In the LABEL method proposed by Sadinlem et al. (2019), the non-conformity score is defined as follows. For each point i of the calibration set:

$$s_i(X_i, Y_i) = 1 - \hat{\mu}(X_i)_{Y_i} \tag{7}$$

This simple approach allows us to construct prediction sets coming with a theoretical guarantee on the marginal coverage. However, although this method generally results in small prediction sets, it tends to produce empty ones when the model is uncertain, for example at the border between two classes.

APS The so-called Adaptive Prediction Set (APS) method overcomes the problem encountered by the LABEL method through the construction of prediction sets which are by definition non-empty (Romano et al., 2020; Angelopoulos et al., 2020). The non-conformity scores are computed by summing the ranked scores of each label, from the higher to the lower until reaching the true label of the observation:

$$s_i(X_i, Y_i) = \sum_{j=1}^{k} \hat{\mu}(X_i)_{\pi_j} \quad \text{where} \quad \pi_k = Y_i \quad \text{and} \quad \hat{\mu}(X_i)_{\pi_1} > ... > \hat{\mu}(X_i)_{\pi_k} > ... > \hat{\mu}(X_i)_{\pi_K} \tag{8}$$

Top-K Introduced by Angelopoulos et al. (2020), the specificity of the Top-K method is that it will give the same prediction set size for all observations. The non-conformity score is the rank of the true label, with scores ranked from higher to lower:

$$s_i(X_i, Y_i) = k \quad \text{where} \quad \pi_k = Y_i \quad \text{and} \quad \hat{\mu}(X_i)_{\pi_1} > ... > \hat{\mu}(X_i)_{\pi_k} > ... > \hat{\mu}(X_i)_{\pi_K} \tag{9}$$

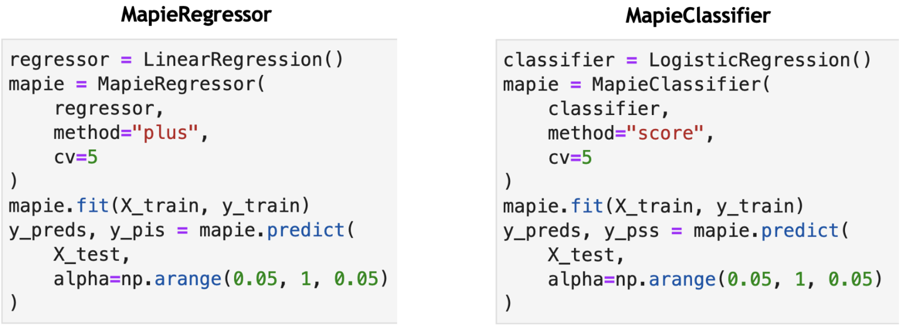
Finally, it should be noted that MAPIE includes split and cross-conformal strategies for the LABEL and APS methods, but only the split-conformal one for Top-K.

RAPS The RAPS method which stands for Regularized Adaptive Prediction Set, is an improvement made by Angelopoulos et al. (2020). This regularization is able to overcome the very large prediction sets given by the APS method.

Intuitively, the goal of this method is to penalize the prediction sets whose sizes are greater than the optimal prediction set size. The level of this regularization is controlled by a hyper-parameter λ. The non-conformity scores are computed by summing the regularized ranked scores of each label, from the higher to the lower until reaching the true label of the observation.

5. MAPIE in practice

The MAPIE library offers the choice between several CP methods: two base classes in Python, called `MapieRegressor` for estimating prediction intervals, `MapieClassifier` for estimating prediction sets, then inherited classes called `MapieQuantileRegressor` for CQR and called `MapieTimeSeriesRegressor` for EnbPI. In addition, the MAPIE offers several non-conformity scores and allows to create new non-conformity scores according to Theorem 1. At the time, MAPIE works with only non-conformity measures defined by means of such function. Figure 1 presents the typical commands needed for quantifying the uncertainties. As the MAPIE base classes are inherited from scikit-learn `BaseEstimator` classes, the API is very intuitive to anyone familiar with scikit-learn and follows a initialization-fit-predict process.



MapieRegressor

```
regressor = LinearRegression()
mapie = MapieRegressor(
    regressor,
    method="plus",
    cv=5
)
mapie.fit(X_train, y_train)
y_preds, y_pis = mapie.predict(
    X_test,
    alpha=np.arange(0.05, 1, 0.05)
)
```

MapieClassifier

```
classifier = LogisticRegression()
mapie = MapieClassifier(
    classifier,
    method="score",
    cv=5
)
mapie.fit(X_train, y_train)
y_preds, y_pss = mapie.predict(
    X_test,
    alpha=np.arange(0.05, 1, 0.05)
)
```

Figure 1: Description of the Python commands needed for estimating prediction intervals and prediction sets with `MapieRegressor` and `MapieClassifier`, respectively.

After initializing the base scikit-learn-compatible base model, one needs to initialize the desired MAPIE class with three main arguments that define the strategy for uncertainty quantification:

- the `"cv"` argument defines the train / calibration set splitting strategy for training the model and calibrating the non-conformity scores. It can be `"prefit"` in the split-conformal case where the base model is already fitted on a given training set while the calibration set given to MAPIE is used directly for calibrating the non-conformity scores. For cross-conformal methods, one can simply define an integer that sets the number of splits and MAPIE will call internally the corresponding `BaseCrossValidator` object, such as `LeaveOneOut` and `KFold` for the Jackknife or CV strategies, respectively.
- the `"method"` argument controls the strategy for constructing the prediction intervals or prediction sets. For regression tasks, one can choose among `"base"`, `"plus"`, or `"minmax"`. For example, `method="plus"` together with `cv=KFold(5)` defines the CV+ method with 5 folds. For classification, one can choose `"score"` for the LABEL method from Sadinlem et al. (2019), `"cumulated_score"` for the Adaptive Prediction Set (APS) method by Romano et al. (2020), and `"top_k"` for the Top-K method by Angelopoulos et al. (2020).

- the "conformity_score" argument defines the non-conformity score used for constructing the prediction intervals only for regression tasks (for classification tasks, the "method" argument constrains the non-conformity score to use). One can choose any instance that inherits the class ConformityScore like AbsoluteConformityScore for the standard absolute residual non-conformity score or GammaConformityScore for the gamma non-conformity score.

Methods that deviate from these standard cases (i.e. training and test exchangeable datasets sampled from similar distribution) need to be implemented in other classes by inheriting from MapieRegressor or MapieClassifier as base classes. For instance, the EnbPI method (Xu and Xie, 2020) or Conformalized Quantile method (Romano et al., 2019) are implemented in MapieTimeSeriesRegressor and MapieQuantileRegressor respectively. They both inherit from the base class MapieRegressor. This process allows anyone to suggest an implementation of a new conformal prediction method through a dedicated pull request that follows the MAPIE guidelines without modifying the base classes.

6. MAPIE by example

Here we propose to show how we can use MAPIE in practice for different purposes. First, we will illustrate in regression tasks how the non-conformity score can impact the width of the prediction intervals and the local coverage. Finally, we present two simple examples of uncertainty quantification on time series and computer vision settings using MAPIE. Notebooks for more examples can be found in the GitHub repository[3].

6.1. MAPIE for regression with non-conformity score functions

We illustrate uncertainty quantification on regression and analyze the impact of the non-conformity score on the width of the prediction intervals and on the local coverage score. Since the distribution of the residuals has an impact on the width of the prediction intervals and thus conditions the choice of the p-normalized residuals non-conformity scores, we propose to analyze their behavior on different datasets:

- the first one is a **toy dataset with homoscedastic noise** $\mathcal{D}_1 = \{(X_i, Y_i)\}_{1\leq i\leq n}$ that is generated based on a linear model as follows: $X \sim \mathcal{U}([0, 100]^m)$ and $Y = \beta^T X + \sigma\epsilon$ with $\beta \in \mathbb{R}^m$, $\epsilon \sim \mathcal{N}(0, 1)$ and $\sigma \in \mathbb{R}^+$ ($m = 10$ is the number of attributes). In other words, the noise is independent of the observation and the target. In this setting, the p-normalized non-conformity score with $p = 0$ should be the best a priori.

- the second one is a **toy dataset with heteroscedastic noise** $\mathcal{D}_2 = \{(X_i, Y_i)\}_{1\leq i\leq n}$ that is generated based on a linear model with target-dependent noise as follows: $X \sim \mathcal{U}([0, 100]^m)$ and $Z = \beta^T X + \epsilon_1$ with $\beta \in \mathbb{R}^m$, $\epsilon_1 \sim \mathcal{N}(0, 1)$, then $Y = Z(1 + \epsilon_2)$ with $\epsilon_2 \sim \mathcal{N}(0, 1)$. In words, the larger the target, the greater the noise. In this setting, the p-normalized non-conformity score with $p = 2$ should be the best a priori because the quantile will be at the prediction scale: $\hat{q}^{\pm}_{n,\alpha}(x) \propto \hat{\mu}(x)$.

[3] Link to access the MAPIE repository: https://github.com/scikit-learn-contrib/MAPIE

- the last one is the **house price dataset**[4] that consists of predicting the final price of a home using 79 explanatory variables describing (almost) all aspects of residential housing in Ames, Iowa. For the sake of simplicity, we have kept only 5 of the 79 variables, such as the type of dwelling involved in the sale, the lot size in square feet, the size of garage in square feet, the rate of the overall material and finish of the house and the rate of the the overall condition of the house.

We performed a linear regression on these datasets. Concerning the MAPIE model used for these regression tasks, the base ML model is a standard linear regression of scikit-learn library (by defining `estimator=LinearRegression`) which we encapsulate in the `MapieRegressor` class. We propose to test different cross-conformal method named CV+ (via the `method='plus'` and `cv='10'` parameters) with different p-normalized non-conformity scores ($p = 0$ that is equivalent to the absolute residual non-conformity score and $p = 2$) and compare them with the CQR split-conformal method. For all experiments, we choose a risk level $\alpha = 0.1$.

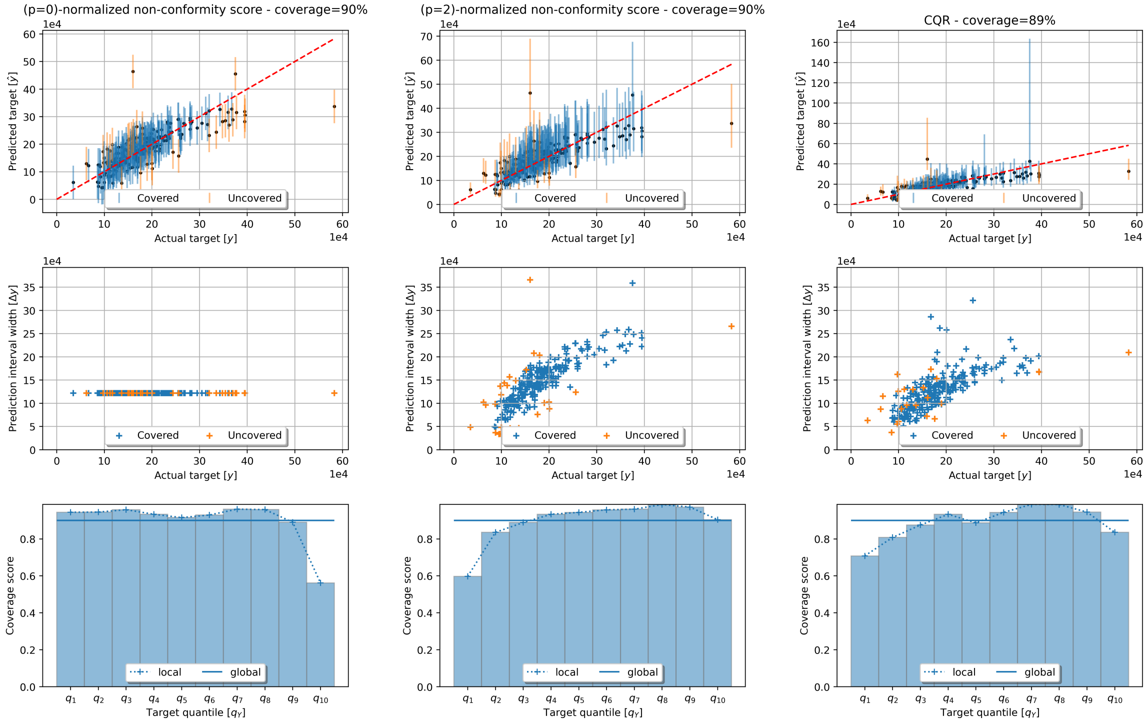
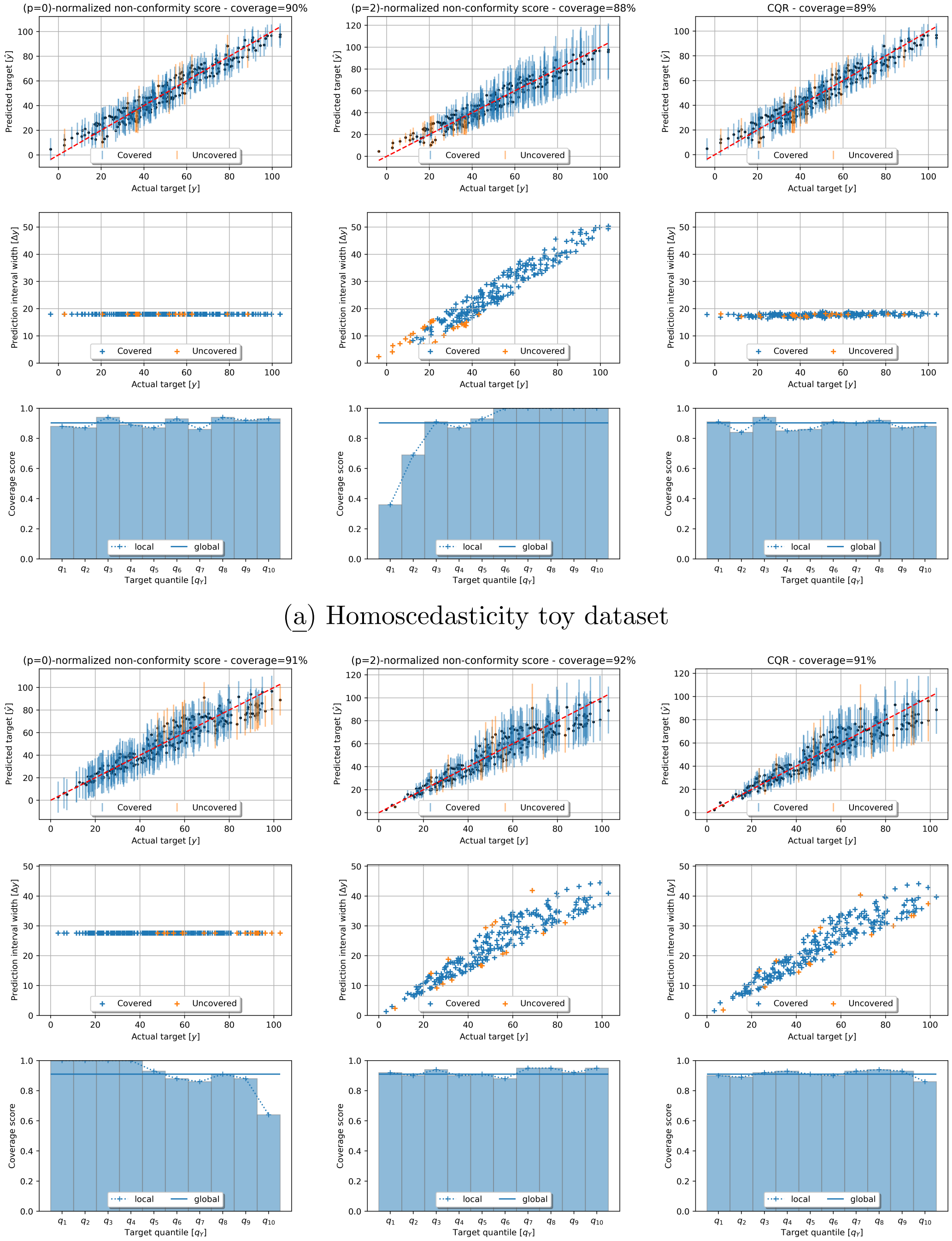


Figure 2: Comparison on the **house prise dataset** of the coverage of the cross-conformal method (CV+) with two different normalized residual non-conformity scores of degree $p = 0$ and $p = 2$ and the CQR method. The based model is a linear regression and the risk level is fixed at $\alpha = 0.1$. In rows: 1) predicted target vs. actual target, 2) width of the prediction interval vs. actual target, and 3) local coverage score vs. target quantile.

[4] Link to access the dataset: https://www.openml.org/search?type=data&status=active&id=42165



(a) Homoscedasticity toy dataset

(b) Heteroscedasticity toy dataset

Figure 3: Comparison of the coverage of the cross-conformal method (CV+) with two different normalized residual non-conformity scores of degree $p = 0$ and $p = 2$ and the CQR method. The risk level is fixed at $\alpha = 0.1$. In rows: (1) predicted target vs. actual target, (2) width of the prediction interval vs. actual target, and (3) local coverage score vs. target quantile.

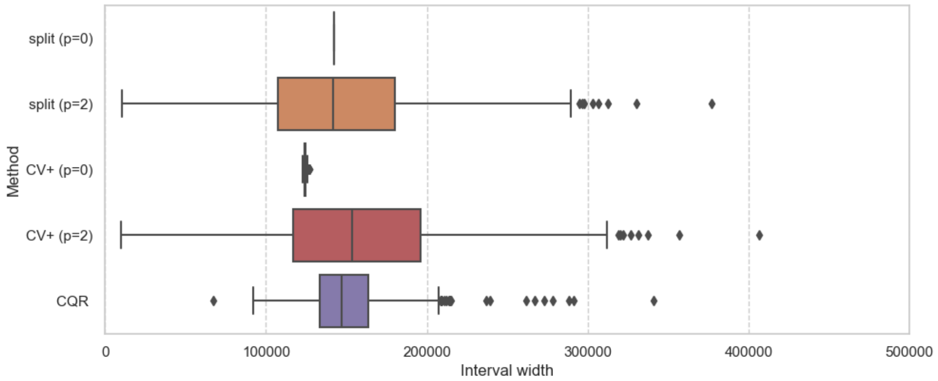


Figure 4: Distribution via boxplot of the width of prediction intervals (in \$) on the house price dataset. `split` is for split-conformal prediction method, `CV+` for cross-conformal prediction method and `CQR` for Conformalized Quantile Regression.

The results obtained are presented in Figure 3(a) for the homoscedastic toy dataset, Figure 3(b) for the heteroscedastic toy dataset and Figure 2 for the house price dataset. Our findings show that the method with absolute residual non-conformal score produces prediction intervals with constant width on homoscedastic dataset, while the method with normalized residual non-conformal score produces prediction intervals whose width increases with the prediction. Furthermore, the latter method favors wide intervals for large values, resulting in too much coverage, and narrow intervals for small values, resulting in too little coverage, while the former guarantees coverage on each quantile bin. On heteroscedastic dataset, the situation is reversed, where the second method achieves good local coverage while the first one does not.

In the case of the house price dataset, we observed that the first and second methods failed to provide accurate prediction intervals for small and large values, respectively. In complement to this observation, we provide in Figure 4 the distribution of the width of prediction intervals to see their variability. However, the CQR method was able to strike a balance between both areas and guarantee local coverage (Figure 2). It is important to note that while the CQR method can manage both heteroscedastic and homoscedastic noise, it is not directly connected to the prediction function $\hat{\mu}$ and therefore requires the re-training or change of the existing model.

In summary, MAPIE is easy-to-use with a wide choice of combinations to produce systematic prediction intervals with marginal coverage guarantee in regression. Our analysis highlights that the choice of the p-normalized non-conformity score depends on the problem and requires an analysis of the distribution of residuals. Additionally, the expert may prefer to over-cover either small or large values of the target, depending on their specific needs. For instance, in the context of estimating house prices, experts may accept not sufficiently covering small estimates but prefer to ensure that the true estimation falls within the prediction interval for large targets.

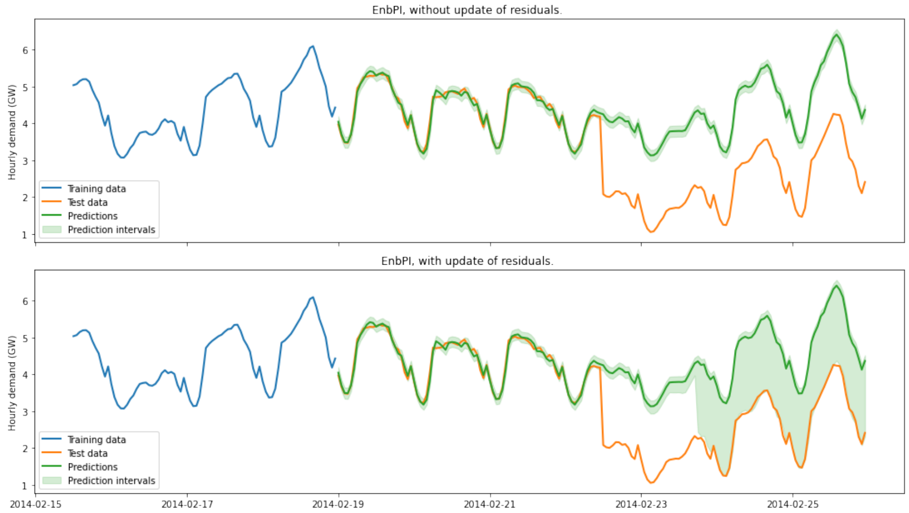


Figure 5: Predictions obtained by `MapieTimeSeriesRegressor` without (top) and with (bottom) partial update of residuals for computing prediction intervals from a Random Forest regressor as based model trained on the Australian electricity dataset.

6.2. MAPIE for Time series

We illustrate the EnbPI algorithm with `MapieTimeSeriesRegressor` on the Victoria electricity demand dataset with an artificial change point on the test set. The Victoria electricity demand dataset consists in the hourly demand of electricity in the Victoria state in Australia between January 1st and February 23rd of 2014. The test set is the last week, and the remaining weeks are included in the training set. We added a sudden decrease of the electricity demand of 2 GW on February 22 to simulate a change point in the test set. The explanatory variables are the lags of the demand up to 5 previous hours and the temperature. The base ML model is a Random Forest whose hyperparameters are optimized through chronological cross-validation. We then use this tuned Random Forest as the base model for the EnbPI method, using 100 block bootstrap resamplings and with a block length of 48 hours. We sequentially estimate the prediction interval on the electricity demand one step ahead.

Figure 5 compares the estimated prediction intervals without and with update of the residuals, that is a key point of the EnbPI method. The training data do not contain a change point, hence the base model cannot anticipate it. Without update of the residuals, the prediction intervals are built upon the distribution of the residuals of the training set. Therefore they do not cover the true observations after the change point, leading to a sudden decrease of the coverage. However, the partial update of the residuals allows the method to capture the increase of uncertainties of the model predictions. One can notice that the uncertainty's explosion happens about one day late. This is because enough new residuals are needed to change the quantiles obtained from the residuals distribution.

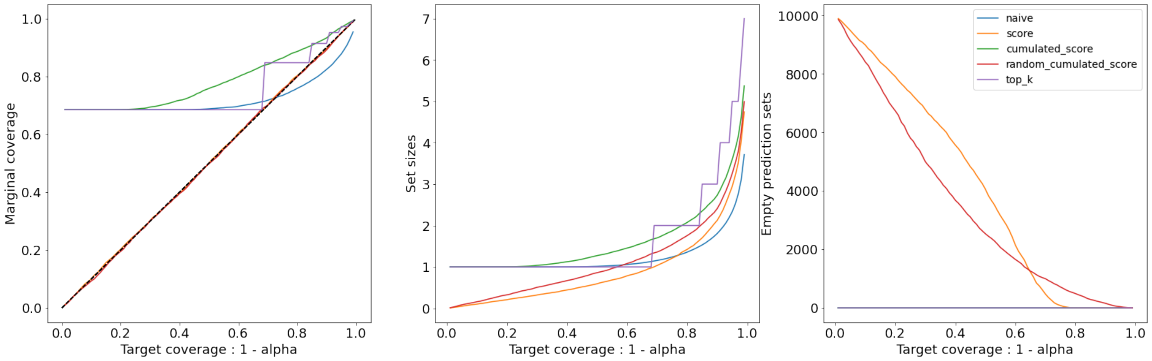


Figure 6: Comparison of the number of empty prediction sets, marginal coverages and average prediction set sizes for the different classification methods implemented in `MapieClassifier`.

6.3. MAPIE for image classification

We illustrate uncertainty quantification on image classification with the famous CIFAR10 dataset which consists in images belonging to 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

As mentioned before, MAPIE is a ”scikit-learn compatible” library, meaning that the base classifier must include the `classes_`, `trained_` attributes and the `fit`, `train`, `predict`, `predict_proba`, `__sklearn_is_fitted__` methods in order to be accepted by `MapieClassifier`. Hence, for computer vision settings, one can create a wrapper around a deep learning model object (or any other non-scikit-learn compatible ML model) to be digested by `MapieClassifier`. An example of such a wrapper can be found in the CIFAR10 notebook of the MAPIE repository[5].

After training a small convolutional neural network on the CIFAR10 dataset, we used this wrapper to fit MAPIE on a calibration set and create prediction sets on a test set. Figure 6 compares the different aforedescribed methods implemented in MAPIE with a naive one which directly includes all the labels such that their sum is just above the target coverage level.

As expected, the `naive` method has a coverage which is below the target coverage for α values higher than 0.6: this is because output scores of the model do not represent probabilities. Thus, even by adding the labels, there are no guarantees that the coverage will be achieved. Other methods, on the other hand, all achieve the required coverage regardless the α value because they calibrate the scores with a dataset not seen by the model during training.

Even though the methods all achieve the required coverages, the averaged sizes of their prediction sets are quite different, especially at low α values. As discussed in the previous section, the `score` method achieves the lowest averaged size for all target coverages compared to the `cumulated_score` and `top_k` methods. However, the `score` method

[5] Link to access the notebook: https://github.com/scikit-learn-contrib/MAPIE/blob/master/notebooks/classification/Cifar10.ipynb

may also result in empty prediction sets. This situation arises in uncertain cases where the predicted scores of all labels do not reach the target quantile. On the second hand, the `cumulated_score`, by definition, always includes at least one label, the one whose `cumulated_score` is higher than the quantile, but induces over-estimated marginal coverages for low α values. To force the marginal coverage to stay close to the target one, one can choose the `random_cumulated_score` which includes randomly the last label. The major drawback of the latter method is that it creates empty prediction sets which indicate low uncertainties, unlike the `score` method.

7. Conclusion and future works

We introduced MAPIE, an open-source library for flexible and systematic uncertainty estimation with marginal coverage guarantee. Our library is easy-to-use and provides a wide choice of combinations for both regression and classification tasks. Through our experiments, we have demonstrated that MAPIE can handle different types of data, including heteroscedastic datasets, while ensuring global coverage for the prediction intervals.

A framework for straightforward implementation of new methods One of the main strengths of MAPIE is its modular and flexible architecture, which allows straightforward implementation of new conformal prediction methods. Our library provides an easy-to-use API that enables researchers and practitioners to quickly integrate their own methods into MAPIE for prototyping and production purposes. This makes it possible to easily compare different methods and choose the most appropriate one for a given problem.

Extension to new conformal prediction paradigms In the landscape of conformal prediction research, some challenges are not yet addressed but must be managed in the future. These include adaptive conformal inference (Gibbs and Candes, 2021), co-variate shift (Tibshirani et al., 2019), multi-target regression through copula-based conformal prediction (Messoudi et al., 2021), binary classification (Vovk et al., 2015) and conformal label shift (Podkopaev and Ramdas, 2021).

Furthermore, we focus in particular on Risk-controlling Prediction Sets (Bates et al., 2021) or the Learn-Then-Test (Angelopoulos et al., 2021) frameworks, which have the potential to improve the performance of prediction intervals in specific applications. We believe that our library will provide a useful tool for researchers and practitioners working in the field of conformal prediction, enabling them to easily test and compare different methods and to develop new ones.

Acknowledgments

We would like to thank Quantmetry, Confiance.ai, Région Ile-de-France for their financial support (AI4industry), as well as the Michelin AI R&D team for enlightening discussions.

References

Anastasios Nikolas Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. ArXiv, abs/2009.14193, 2020. doi: 10.48550/ARXIV.2009.14193.

Anastasios Nikolas Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. ArXiv, abs/2110.01052, 2021.

S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. Jordan. Distribution-free, risk-controlling prediction sets. Journal of the ACM (JACM), 68(6):1–34, 2021.

Gianluca Detommaso, Alberto Gasparin, Michele Donini, Matthias W. Seeger, Andrew Gordon Wilson, and C. Archambeau. Fortuna: A library for uncertainty quantification in deep learning. ArXiv, abs/2302.04019, 2023.

B. Efron. Bootstrap methods: Another look at the jackknife. The Annals of Statistics, 7 (1):1–26, 1979. ISSN 00905364. URL http://www.jstor.org/stable/2958830.

R. Foygel Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. Ann. Statist., 49(1):486–507, 02 2021. doi: 10.1214/20-AOS1965. URL https://doi.org/10.1214/20-AOS1965.

Soumya Shubhra Ghosh, Qingzi Vera Liao, Karthikeyan Natesan Ramamurthy, Jirí Navrátil, Prasanna Sattigeri, Kush R. Varshney, and Yunfeng Zhang. Uncertainty quantification 360: A holistic toolkit for quantifying and communicating the uncertainty of ai. ArXiv, abs/2106.01410, 2021.

I. Gibbs and E. Candes. Adaptive conformal inference under distribution shift. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 1660–1672. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/0d441de75945e5acbc865406fc9a2559-Paper.pdf.

B. Kim, C. Xu, and Rina Barber. Predictive inference is free with the jackknife+-after-bootstrap. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 4138–4149. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/2b346a0aa375a07f5a90a344a61416c4-Paper.pdf.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. Journal of the American Statistical Association, 113(523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116. URL https://doi.org/10.1080/01621459.2017.1307116.

Henrik Linusson. nonconformist. https://github.com/donlnz/nonconformist, 2022. Accessed: 2023-04-03.

S. Messoudi, Sébastien Destercke, and Sylvain Rousseau. Copula-based conformal prediction for multi-target regression. Pattern Recognit., 120:108101, 2021.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, Machine Learning: ECML 2002, pages 345–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-36755-0.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In Conference on Uncertainty in Artificial Intelligence, 2021.

M. H. Quenouille. Notes on bias in estimation. Biometrika, 43(3/4):353–360, 1956. ISSN 00063444. URL http://www.jstor.org/stable/2332914.

Y. Romano, E. Patterson, and E. Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf.

Y. Romano, M. Sesia, and E. Candes. Classification with valid and adaptive coverage. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 3581–3591. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/244edd7e85dc81602b7615cd705545f5-Paper.pdf.

M. Sadinlem, J. Lei, and L. Wasserman. Least ambiguous set-valued classifiers with bounded error levels. Journal of the American Statistical Association, 114(525):223–234, 2019. doi: 10.1080/01621459.2017.1395341. URL https://doi.org/10.1080/01621459.2017.1395341.

Matteo Sesia and Emmanuel J Candès. A comparison of some conformal quantile regression methods. Stat, 9(1):e261, 2020.

Martim Sousa, Ana Maria Tom’e, and Jos’e Moreira. Improved conformalized quantile regression. ArXiv, abs/2207.02808, 2022.

Vianney Taquet, V. Blot, Thomas Morzadec, Louis Lacombe, and Nicolas J.-B. Brunel. Mapie: an open-source library for distribution-free uncertainty quantification. ArXiv, abs/2207.12274, 2022.

Ryan Tibshirani. Conformal inference r project. https://github.com/ryantibs/conformal, 2022. Accessed: 2023-04-03.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In Neural Information Processing Systems, 2019.

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. Advances in Neural Information Processing Systems, 28, 2015.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In International Conference on Machine Learning, 2020.

Appendix A. Details of cross-conformal methods in MAPIE

MAPIE has three state-of-the-art conformal prediction methods (along with their derivatives) for tabular regression : Jackknife+, CV+ and Jackknife+-after-Bootstrap.

Jackknife+ Presented in Foygel Barber et al. (2021), the Jackknife+ method is based on the standard Jackknife, which constructs a set of *leave-one-out* models. Estimating the prediction intervals is carried out in three main steps. First, train n leave-one-out models (one for each training instance), each leave-one-out model is trained on the entire training set except the corresponding training point. Second, compute the corresponding conformity scores (here, the leave-one-out residuals) $|Y_i - \hat{\mu}_{-i}(X_i)|$. Third, fit the regression function $\hat{\mu}$ on the entire training set and construct the prediction intervals from the distribution of the computed leave-one-out residuals and the desired $1-\alpha$ quantile. This method avoids over-fitting but still does not guarantee the targeted coverage if $\hat{\mu}$ is unstable, for example when the sample size is close to the number of features (Foygel Barber et al., 2021).

Unlike the standard jackknife method which returns a prediction interval centered around the prediction of the model trained on the entire dataset, the so-called Jackknife+ method also uses the predictions from all leave-one-out models on the new test point to take the variability of the regression function into account. The resulting confidence interval can therefore be summarized as follows

$$\hat{C}_{n,\alpha}^{\text{jackknife+}}(X_{n+1}) = [\hat{q}_{n,\alpha}^{-}\{\hat{\mu}_{-i}(X_{n+1}) - R_i^{\text{LOO}}\}, \hat{q}_{n,\alpha}^{+}\{\hat{\mu}_{-i}(X_{n+1}) + R_i^{\text{LOO}}\}] \tag{10}$$

As described in Foygel Barber et al. (2021), this method guarantees a higher stability with a coverage level of $1-2\alpha$ for a target coverage level of $1-\alpha$, under the assumption of independence of the distribution of the data (X, Y).

CV+ In order to reduce the computational cost, one can adopt a cross-validation approach instead of a leave-one-out approach, with the so-called CV+ method. Similar to the Jackknife+ method, estimating the prediction intervals with CV+ is performed in four main steps. First, split the training set into K disjoint subsets $S_1, S_2, ..., S_K$ of equal size. Second, regression functions $\hat{\mu}_{-S_k}$ are fitted on the training set with the corresponding k^{th} fold removed. Third, the corresponding out-of-fold residuals are computed for each i^{th} point $|Y_i - \hat{\mu}_{-S_{k(i)}}(X_i)|$ where $k(i)$ is the fold containing i. Finally, similar to Jackknife+, the regression functions $\hat{\mu}_{-S_{k(i)}}(X_i)$ are used to estimate the prediction intervals.

As for Jackknife+, this method guarantees a coverage level higher than $1-2\alpha$ for a target coverage level of $1-\alpha$, assuming essentially the independence of the data. As pointed out by Foygel Barber et al. (2021), Jackknife+ can be considered as a special case of CV+ with $K = n$. In practice, this method results in slightly wider prediction intervals and is therefore likely more conservative, but gives a reasonable compromise for large datasets when the Jackknife+ method is unfeasible.

Jackknife+-after-bootstrap An alternative way to reduce the computational cost is to adopt a bootstrap approach instead of cross-validation, called the Jackknife+-after-bootstrap method, offered by Kim et al. (2020). Similar to CV+, estimating the prediction intervals with Jackknife+-after-bootstrap is performed in four main steps. First, resample the training set with replacement (bootstrap) K times, to get the (non disjoint) bootstraps $B_1, ..., B_K$ of equal size. Second, fit K regressions functions $\hat{\mu}_{B_k}$ on the bootstraps (B_k), and compute the predictions on the complementary sets B_k^c. Third, aggregate these predictions according to a given aggregation function, typically mean or median, and compute the residuals $|Y_j - \text{agg}(\hat{\mu}(B_{K(j)}(X_j)))|$ are computed for each X_j (with $K(j)$ the bootstraps not containing X_j). The sets $\{\text{agg}(\hat{\mu}_{K(j)}(X_i) + r_j\}$ (where j indexes the training set) are used to estimate the prediction intervals.

As for Jackknife+, this distribution-free method guarantees a coverage level higher than $1 - 2\alpha$ for a target coverage level of $1 - \alpha$.

Appendix B. Details of the non-conformity scores used for classification in MAPIE

Three methods for multi-class classification UQ have been implemented in MAPIE so far: LABEL (Sadinlem et al., 2019), Adaptive Prediction Sets (Romano et al., 2020) and Top-K (Angelopoulos et al., 2020). The difference between these methods is the way the non-conformity scores are computed.

LABEL In the LABEL method, the non-conformity score is defined as as one minus the score of the true label. For each point i of the calibration set:

$$s_i(X_i, Y_i) = 1 - \hat{\mu}(X_i)_{Y_i} \tag{11}$$

Once the non-conformity scores $s_1, ..., s_n$ are estimated for all calibration points, we compute the $(n + 1) * (1 - \alpha)/n$ quantile $\hat{q}$ as follows:

$$\hat{q} = Quantile\left(s_1, ..., s_n; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right) \tag{12}$$

Finally, we construct a prediction set by including all labels with a score higher than the estimated quantile:

$$\hat{C}(X_{test}) = \{y : \hat{\mu}(X_{test})_y \geq 1 - \hat{q}\}$$

This simple approach allows us to construct prediction sets coming with a theoretical guarantee on the marginal coverage. However, although this method generally results in small prediction sets, it tends to produce empty ones when the model is uncertain, for example at the border between two classes.

APS The so-called Adaptive Prediction Set (APS) method overcomes the problem encountered by the LABEL method through the construction of prediction sets which are by definition non-empty. The non-conformity scores are computed by summing the ranked scores of each label, from the higher to the lower until reaching the true label of the observation:

$$s_i(X_i, Y_i) = \sum_{j=1}^{k} \hat{\mu}(X_i)_{\pi_j} \quad \text{where} \quad \pi_k = Y_i \quad \text{and} \quad \hat{\mu}(X_i)_{\pi_1} > ... > \hat{\mu}(X_i)_{\pi_k} > ... > \hat{\mu}(X_i)_{\pi_n}$$

The quantile $\hat{q}$ is then computed the same way as the score method. For the construction of the prediction sets for a new test point, the same procedure of ranked summing is applied until reaching the quantile, as described in the following equation:

$$\hat{C}(X_{test}) = \{\pi_1, ..., \pi_k\} \quad \text{where} \quad k = \inf\{k : \sum_{j=1}^{k} \hat{\mu}(X_{test})_{\pi_j} \geq \hat{q}\}$$

By default, the label whose cumulative score is above the quantile is included in the prediction set. However, its incorporation can also be chosen randomly based on the difference between its cumulative score and the quantile so the effective coverage remains close to the target (marginal) coverage. We refer the reader to Romano et al. (2020) and Angelopoulos et al. (2020) for more details about this aspect.

Top-K Introduced by Angelopoulos et al. (2020), the specificity of the Top-K method is that it will give the same prediction set size for all observations. The non-conformity score is the rank of the true label, with scores ranked from higher to lower. The prediction sets are build by taking the $\hat{q}^{th}$ higher scores. The procedure is described in equation 13.

$$s_i(X_i, Y_i) = k \quad \text{where} \quad \pi_k = Y_i \quad \text{and} \quad \hat{\mu}(X_i)_{\pi_1} > ... > \hat{\mu}(X_i)_{\pi_k} > ... > \hat{\mu}(X_i)_{\pi_n}$$

$$\hat{q} = \left\lceil Quantile\left(s_1, ..., s_n; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right) \right\rceil \tag{13}$$

$$\hat{C}(X_{test}) = \{\pi_1, ..., \pi_{\hat{q}}\}$$

Finally, it should be noted that MAPIE includes split- and cross-conformal strategies for the LABEL and APS methods, but only the split-conformal one for Top-K.

RAPS The RAPS method which stands for Regularized Adaptive Prediction Set, is an improvement made by Angelopoulos et al. (2020). This regularization is able to overcome the very large prediction sets given by the APS method. The non-conformity scores are computed by summing the regularized ranked scores of each label, from the higher to the lower until reaching the true label of the observation:

$$s_i(X_i, Y_i) = \sum_{j=1}^{k} \hat{\mu}(X_i)_{\pi_j} + \lambda(k - k_{reg})^{+} \quad \text{where} \quad \pi_k = Y_i \tag{14}$$

Where:

- π_i is the label associated to the i^{th} score sorted in descending order.
- $(z)^{+}$ denotes the positive part of z

- k_{reg} is the optimal set size (in the sense that if all prediction sets have k_{reg} elements, then one achieves the desired coverage)
- λ is a regularization to be selected according to a methodology explained hereafter.

The optimization of k_{reg} and λ requires an extra data-splitting (by default, 20% of the calibration data). To choose k_{reg}, we simply run the Top-K method over this new split. For the choice of λ, we follow the guidelines of Angelopoulos et al. (2020) and try to find the value of lambda such that it minimizes the size of the prediction set. A simple grid search if done on different values of λ (to be consistent with the original paper, we choose $\lambda \in \{0.001, 0.01, 0.1, 0.2, 0.5\}$).

For the construction of the prediction set for a new test point, the following procedure is applied:

$$\hat{C}(X_{test}) = \{\pi_1, ..., \pi_k\} \quad \text{where} \quad k = \inf\{k : \sum_{j=1}^{k} \hat{\mu}(X_{test})_{\pi_j} + \lambda(k - k_{reg})^+ \geq \hat{q}\} \tag{15}$$

Intuitively, the goal of the method is to penalize the prediction sets whose sizes are greater than the optimal prediction set size. The level of this regularization is controlled by the parameter λ.

Despite that RAPS methods have a relatively small set size, its coverage tends to be higher than the one required (especially for high values of α, which means a low level of confidence). Hence, to achieve exact coverage, one can implement randomization concerning the inclusion of the last label in the prediction set. This randomization is done as follows:

- First: define the V parameter:

$$V_i = (s_i(X_i, Y_i) - \hat{q}_{1-\alpha}) / \left(\hat{\mu}(X_i)_{\pi_k} + \lambda \mathbb{1}(k > k_{reg})\right)$$

- Compare each V_i to $U \sim$ Unif(0, 1)
- If $V_i \leq U$, the last included label is removed, else we keep the prediction set as it is.

Appendix C. Proof of Theorem 1

Non-conformity scores and prediction function

In order to generalize the coverage guarantee to more non-conformity scores (not only the absolute values of residuals), let's first define two functions with some properties. The *signed non-conformity score function* f defined on E^2 ($E \subset \mathbb{R}$) and the *perturbed prediction function* g defined on $F \times E$ ($F \subset \mathbb{R}$). Uppercase letters are used to follow the paper notation Foygel Barber et al. (2021).

Non-conformity scores and prediction function

From an observation and a prediction, the function f outputs the conformity score:

$$\begin{aligned} f : E^2 &\to F \\ (Y, \hat{Y}) &\mapsto f(Y, \hat{Y}) \end{aligned} \tag{16}$$

It verifies the following properties with $S_{k(i)}$ being the subset containing datapoint i (which may be a subset of a single point, for instance in the jackknife method):

$$\begin{aligned} &\forall Y_l < Y_h : f(Y_l, \hat{Y}) < f(Y_h, \hat{Y}) \\ &\forall \hat{Y}_l < \hat{Y}_h : f(Y, \hat{Y}_l) > f(Y, \hat{Y}_h) \\ &\forall i \neq j : R_{i,j} = f(Y_i, \hat{\mu}_{-(S_{k(i)}, S_{k(j)})}(X_i)) \end{aligned} \tag{17}$$

Prediction function

From a conformity score and a prediction, the function g outputs a perturbed prediction:

$$\begin{aligned} g : F \times E &\to E \\ (R, \hat{Y}) &\mapsto g(R, \hat{Y}) \end{aligned} \tag{18}$$

It verifies the following properties:

$$\begin{aligned} &\forall R_l < R_h : g(R_l, \hat{Y}) < g(R_h, \hat{Y}) \\ &\forall i \neq j : Y_i = g(R_{i,j}, \hat{\mu}_{-(S_{k(i)}, S_{k(j)})}(X_i)) \end{aligned} \tag{19}$$

Function relationship

The two functions must verify the following property:

$$\forall \hat{Y}, \forall R : R = f(g(R, \hat{Y}), \hat{Y}) \tag{20}$$

The jackknife+ prediction interval

Let's derive the proof of the theorem to build "the jackknife+ prediction interval" (Foygel Barber et al., 2021) (Theorem 1, part 6, p. 18-21) with more general non-conformity scores.

Construction of R

Instead of defining the R matrix with the absolute value of the residuals, let's assume that R is defined thanks to a non-conformity score f, with $R_{i,j}$ being the non-conformity score of the point i using a model fitted on the training plus test data, with points i and j removed:

$$\begin{aligned} R_{i,j} &= f(Y_i, \hat{\mu}_{-(i,j)}(X_i)) \\ R_i^{LOO} &= R_{i,n+1} \end{aligned} \tag{21}$$

Construction of A

According to the paper, comparison matrices A can be built. $A_{i,j}$ is the indicator for the event that, when excluding data points i and j from the regression, data point i has a more extreme (may be higher or lower depending on the considered case, as described below) non-conformity score than data point j. There are two cases:

1. the symmetrical case when the conformity scores are assumed to be symmetrical around 0. In this case, the absolute values of the conformity scores are used to construct the A matrix.
2. otherwise, the signed conformity scores are kept as they are.

Symmetrical case

According to the paper, the comparison A matrix is defined as below:

$$A_{i,j} = \mathbb{1}\{|R_{i,j}| > |R_{j,i}|\}$$

Asymmetrical case

The two following definitions of comparison matrix A are considered:

$$\begin{aligned} A^{+}_{i,j} &= \mathbb{1}\{R_{i,j} > R_{j,i}\} \\ A^{-}_{i,j} &= \mathbb{1}\{R_{i,j} < R_{j,i}\} \end{aligned} \tag{22}$$

Strange points

As above, the definition of strange points depends on the case considered.

Symmetrical case

According to the paper, there is a single set $S(A) = \{i \in \{1, ..., n+1\} : A_{i.} \geq (1-\alpha)(n+1)\}$.

Asymmetrical case

Instead of the single set $S(A)$, two sets should be considered: $S(A^{+})$ and $S(A^{-})$. Keeping the same definition of the set S of the paper, we have:

1. $S_{+} = S(A^{+}) = \{i \in \{1, ..., n+1\} : A^{+}_{i.} \geq (1 - \frac{\alpha}{2})(n+1)\}$ for "data points with unusually large [positive] residuals", and
2. $S_{-} = S(A^{-}) = \{i \in \{1, ..., n+1\} : A^{-}_{i.} \geq (1-\frac{\alpha}{2})(n+1) \iff (n-A^{+}_{i.}) \geq (1-\frac{\alpha}{2})(n+1)\}$ for "data points with unusually large [negative] residuals".

Step 1 and 2

Changing R and A, does not change the results of steps 1 and 2 from the proof of the original paper (Foygel Barber et al., 2021) (Theorem 1, part 6, p. 18-21) considering absolute value of the residuals. From step 1, in the symetrical case, we get $|S(A)| \leq 2\alpha(n+1)$. And in the asymmetrical case $|S_+| \leq \alpha(n+1)$ and $|S_-| \leq \alpha(n+1)$ since the new matrices have the same property as in the paper. From step 2, we have that the probability of the test point $n+1$ being strange inside S_+ is bounded by α and the probability of the test point $n+1$ being strange inside S_- is also bounded by α for the asymmetrical case (and the probability of the test point $n+1$ being strange inside $S(A)$ is bounded by 2α for the symmetrical case).

Step 3

Symmetrical case

The prediction interval $\hat{C}^{jackknife+}_{n,\alpha}(X_{n+1})$ is defined as follows:

$$\hat{C}^{jackknife+}_{n,\alpha}(X_{n+1}) = [\hat{q}^{-}_{n,\alpha}\{g(-|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))\}, \hat{q}^{+}_{n,\alpha}\{g(|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))\}]$$

Similarly to the theorem's proof, let's suppose that $Y_{n+1} \notin \hat{C}^{jackknife+}_{n,\alpha}(X_{n+1})$. We remind that $\hat{q}^{+}_{n,\alpha}\{v_i\} =$ the $\lceil (1-\alpha)(n+1) \rceil$ -th smallest value of $v_1, ..., v_n$ and $\hat{q}^{-}_{n,\alpha}\{v_i\} =$ the $\lfloor \alpha(n+1) \rfloor$ -th smallest value of $v_1, ..., v_n$. This means either:

$$Y_{n+1} > \hat{q}^{+}_{n,\alpha}\{g(|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))\}$$

which implies that $Y_{n+1} > g(|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))$ for at least $(1-\alpha)(n+1)$ many indices $j \in \{1, ..., n\}$. Thanks to the properties of f and g we have:

$$\begin{aligned}(1-\alpha)(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} > g(|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \hat{\mu}_{-j}(X_{n+1})) > f(g(|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{f(g(R_{n+1,j}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1})) > |R^{LOO}_j|\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{R_{n+1,j} > |R^{LOO}_j|\} \\ &\leq \sum_{j=1}^{n} \mathbb{1}\{|R_{n+1,j}| > |R^{LOO}_j|\} = \sum_{j=1}^{n} A_{n+1,j} \end{aligned} \tag{23}$$

Or otherwise:

$$Y_{n+1} < \hat{q}^{-}_{n,\alpha}\{g(-|R^{LOO}_j|, \hat{\mu}_{-j}(X_{n+1}))\} \tag{24}$$

which implies that $Y_{n+1} < g(-|R_j^{LOO}|, \hat{\mu}_{-j}(X_{n+1}))$ for at least $(1-\alpha)(n+1)$ many indices $j \in \{1, ..., n\}$.

$$\begin{aligned}
(1-\alpha)(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} < g(-|R_j^{LOO}|, \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \hat{\mu}_{-j}(X_{n+1})) < f(g(-|R_j^{LOO}|, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(g(R_{n+1,j}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1})) < -|R_j^{LOO}|\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{R_{n+1,j} < -|R_j^{LOO}|\} \\
&\leq \sum_{j=1}^{n} \mathbb{1}\{|R_{n+1,j}| > |R_j^{LOO}|\} = \sum_{j=1}^{n} A_{n+1,j} \qquad (25)
\end{aligned}$$

Therefore, in either case, $n+1 \in S(A)$, that is, point $n+1$ is a strange point with high absolute conformity scores.

Thus:

$$\mathbb{P}\{Y_{n+1} \notin \hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})\} \leq 2\alpha$$

and:

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})\} \geq 1 - 2\alpha$$

Asymmetrical case

In this section, the prediction interval $\hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})$ is defined as follows:

$$\hat{C}_{n,\alpha}^{jackknife+}(X_{n+1}) = [\hat{q}_{n,\frac{\alpha}{2}}^{-}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\}, \hat{q}_{n,\frac{\alpha}{2}}^{+}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\}]$$

Similarly to the theorem's proof, let's suppose that $Y_{n+1} \notin \hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})$. This means either:

$$Y_{n+1} > \hat{q}_{n,\frac{\alpha}{2}}^{+}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\}$$

which implies that $Y_{n+1} > g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))$ for at least $(1-\frac{\alpha}{2})(n+1)$ many indices $j \in \{1, ..., n\}$, or otherwise:

$$Y_{n+1} < \hat{q}_{n,\frac{\alpha}{2}}^{-}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\}$$

which implies that $Y_{n+1} < g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))$ for at least $(1-\frac{\alpha}{2})(n+1)$ many indices $j \in \{1, ..., n\}$.

Here R_j^{LOO} can be positive or negative so it is added in both cases. Besides, $\frac{\alpha}{2}$ is used instead of α since to get the $(1-\alpha)$ confidence interval, we look for an upper bound on the confidence interval so that $\frac{\alpha}{2}$ of the residuals are above and a lower bound on the confidence interval so that $\frac{\alpha}{2}$ of the residuals are below.

Let's consider the first case $Y_{n+1} > \hat{q}^{+}_{n,\frac{\alpha}{2}}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\}$. Then, thanks to the properties of f and g:

$$\begin{aligned}
(1-\frac{\alpha}{2})(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} > g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \hat{\mu}_{-j}(X_{n+1})) > f(g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(g(R_{n+1,j}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1})) > R_j^{LOO}\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{R_{n+1,j} > R_{j,n+1}\} = \sum_{j=1}^{n} A^{+}_{n+1,j} \qquad (26)
\end{aligned}$$

and therefore $n+1 \in S_+$, that is, point $n+1$ is a strange point with high positive residuals.

Similarly, we can conclude the same for strange point with high negative residuals.

$$\begin{aligned}
(1-\frac{\alpha}{2})(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} < g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \hat{\mu}_{-j}(X_{n+1})) < f(g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(g(R_{n+1,j}, \hat{\mu}_{-j}(X_{n+1})), \hat{\mu}_{-j}(X_{n+1})) < R_j^{LOO}\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{R_{n+1,j} < R_{j,n+1}\} = \sum_{j=1}^{n} A^{-}_{n+1,j} = n - \sum_{j=1}^{n} A^{+}_{n+1,j} \qquad (27)
\end{aligned}$$

and therefore $n+1 \in S_-$, that is, point $n+1$ is a strange point with high negative residuals.

Combining with step 2, we have:

$$\begin{aligned}
\mathbb{P}\{Y_{n+1} > \hat{q}^{+}_{n,\frac{\alpha}{2}}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \leq \alpha \\
\mathbb{P}\{Y_{n+1} < \hat{q}^{-}_{n,\frac{\alpha}{2}}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \leq \alpha
\end{aligned} \qquad (28)$$

Since any point cannot be simultaneously in S_+ and S_- otherwise $\alpha \geq 1$ (because $A^{+}_{i.} \geq (1-\frac{\alpha}{2})(n+1)$ and $(n - A^{+}_{i.}) \geq (1-\frac{\alpha}{2})(n+1) \Rightarrow n \geq 2(1-\frac{\alpha}{2})(n+1) \iff \alpha \geq \frac{n+2}{n+1}$), a point is at most in one of the two sets.

$$\begin{aligned}
\mathbb{P}\{Y_{n+1} \in \hat{C}^{jackknife+}_{n,\alpha}(X_{n+1})\} = 1 & \\
& - \mathbb{P}\{Y_{n+1} > \hat{q}^{+}_{n,\frac{\alpha}{2}}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \\
& - \mathbb{P}\{Y_{n+1} < \hat{q}^{-}_{n,\frac{\alpha}{2}}\{g(R_j^{LOO}, \hat{\mu}_{-j}(X_{n+1}))\} \\
& \geq 1 - 2\alpha \qquad (29)
\end{aligned}$$

The jackknife-minmax prediction interval

Let's derive the proof of the theorem to build "the jackknife-minmax prediction interval" (Foygel Barber et al., 2021) (Theorem 3, part B.1, p. 31-32) with more general non-conformity scores.

Construction of R

As above, instead of defining the R matrix with the absolute residual, let assume that R is defined thanks to a non-conformity score:

$$\begin{aligned} R_{i,j} &= f(Y_i, \hat{\mu}_{-(i,j)}(X_i)) \\ R_i^{LOO} &= R_{i,n+1} \end{aligned} \tag{30}$$

Construction of A

As above, there are two cases:

1. the symmetrical case when the conformity scores are assumed to be symmetrical around 0. In this case, the absolute values of the conformity scores are used to construct the A matrix.

2. otherwise, the signed conformity scores are kept as they are.

Symmetrical case

According to the paper, the A matrix is defined as below:

$$A_{i,j} = \mathbb{1}\{\min_{j'} |R_{i,j'}| > |R_{j,i}|\}$$

Asymmetrical case

The two following definitions of matrix A are considered:

$$\begin{aligned} A^+_{i,j} &= \mathbb{1}\{\min_{j'} R_{i,j'} > R_{j,i}\} \\ A^-_{i,j} &= \mathbb{1}\{\max_{j'} R_{i,j'} < R_{j,i}\} = \mathbb{1}\{\min_{j'} - R_{i,j'} > -R_{j,i}\} \end{aligned} \tag{31}$$

Strange points

As above, the definition of strange points depends on the case considered.

Symmetrical case

According to the paper, there is a single $S(A)$ set $S(A) = \{i \in \{1, ..., n+1\} : A_{i.} \geq (1-\alpha)(n+1)\}$.

Asymmetrical case

Instead of the single set $S(A)$, two sets should be considered: A^+ and A^-. Keeping the same definition of the set S of the paper, we have:

1. $S_+ = S(A^+) = \{i \in \{1, ..., n+1\} : A^+_{i.} \geq (1 - \frac{\alpha}{2})(n+1)\}$ for "data points with unusually large [positive] residuals", and

2. $S_- = S(A^-) = \{i \in \{1, ..., n+1\} : A^-_{i.} \geq (1 - \frac{\alpha}{2})(n+1) \iff (n - A^+_{i.}) \geq (1 - \frac{\alpha}{2})(n+1)\}$ for "data points with unusually large [negative] residuals".

Step 1 and 2

Changing R and A, does not change the results of steps 1 and 2 from the proof of the original paper (Foygel Barber et al., 2021) (Theorem 1, part 6, p. 18-21) considering absolute value of the residuals. From step 1, in the symetrical case, we get $|S(A)| \leq \alpha(n+1)$. And in the asymmetrical case $|S_+| \leq \frac{\alpha}{2}(n+1)$ and $|S_-| \leq \frac{\alpha}{2}(n+1)$ since the new matrices have the same property as in the paper. From step 2, we have that the probability of the test point $n+1$ being strange inside S_+ is bounded by $\frac{\alpha}{2}$ and the probability of the test point $n+1$ being strange inside S_- is also bounded by $\frac{\alpha}{2}$ for the asymmetrical case (and the probability of the test point $n+1$ being strange inside $S(A)$ is bounded by α for the symmetrical case).

Step 3

Symmetrical case

The prediction interval $\hat{C}^{jack-mm}_{n,\alpha}(X_{n+1})$ is defined as follows:

$$\hat{C}^{jack-mm}_{n,\alpha}(X_{n+1}) = [g(\hat{q}^-_{n,\alpha}\{-|R^{LOO}_j|\}, \min_i \hat{\mu}_{-i}(X_{n+1})), g(\hat{q}_{n,\alpha^+}\{|R^{LOO}_j|\}, \max_i \hat{\mu}_{-i}(X_{n+1}))]$$

Similarly to the theorem's proof, let's suppose that $Y_{n+1} \notin \hat{C}^{jack-mm}_{n,\alpha}(X_{n+1})$. This means either:

$$Y_{n+1} > g(\hat{q}_{n,\alpha^+}\{|R^{LOO}_j|\}, \max_i \hat{\mu}_{-i}(X_{n+1}))$$

which implies that $Y_{n+1} > g(|R_j^{LOO}|, \max_i \hat{\mu}_{-i}(X_{n+1}))$ for at least $(1-\alpha)(n+1)$ many indices $j \in \{1, ..., n\}$ since g is an increasing function regarding the first variable.

$$\begin{aligned}(1-\alpha)(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} > g(|R_j^{LOO}|, \max_i \hat{\mu}_{-i}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \max_i \hat{\mu}_{-i}(X_{n+1})) > f(g(|R_j^{LOO}|, \max_i \hat{\mu}_{-i}(X_{n+1})), \max_i \hat{\mu}_{-i}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{\min_i f(Y_{n+1}, \hat{\mu}_{-i}(X_{n+1})) = f(Y_{n+1}, \max_i \hat{\mu}_{-i}(X_{n+1})) > |R_j^{LOO}|\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{\min_i R_{n+1,i} > |R_j^{LOO}|\} \\ &\leq \sum_{j=1}^{n} \mathbb{1}\{\min_i |R_{n+1,i}| > |R_j^{LOO}|\} = \sum_{j=1}^{n} A_{n+1,j} \qquad (32)\end{aligned}$$

Or otherwise:

$$Y_{n+1} < g(\hat{q}_{n,\alpha}^{-}\{-|R_j^{LOO}|\}, \min_i \hat{\mu}_{-i}(X_{n+1}))$$

which implies that $Y_{n+1} < g(-|R_j^{LOO}|, \min_i \hat{\mu}_{-i}(X_{n+1}))$ for at least $(1-\alpha)(n+1)$ many indices $j \in \{1, ..., n\}$.

$$\begin{aligned}(1-\alpha)(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} < g(-|R_j^{LOO}|, \min_i \hat{\mu}_{-i}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \min_i \hat{\mu}_{-i}(X_{n+1})) < f(g(-|R_j^{LOO}|, \min_i \hat{\mu}_{-i}(X_{n+1})), \min_i \hat{\mu}_{-i}(X_{n+1}))\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{\max_i f(Y_{n+1}, \hat{\mu}_{-i}(X_{n+1})) = f(Y_{n+1}, \min_i \hat{\mu}_{-i}(X_{n+1})) < -|R_j^{LOO}|\} \\ &= \sum_{j=1}^{n} \mathbb{1}\{\max_i R_{n+1,i} < -|R_j^{LOO}|\} \\ &\leq \sum_{j=1}^{n} \mathbb{1}\{-\min_i |R_{n+1,i}| < -|R_j^{LOO}|\} = \sum_{j=1}^{n} A_{n+1,j} \qquad (33)\end{aligned}$$

Therefore, in either case, $n+1 \in S(A)$, that is, point $n+1$ is a strange point with high absolute conformity scores.

Thus:

$$\mathbb{P}\{Y_{n+1} \notin \hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})\} \leq \alpha$$

and:

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{n,\alpha}^{jackknife+}(X_{n+1})\} \geq 1-\alpha$$

Asymmetrical case

In this section, the prediction interval is defined as follows:

$$\hat{C}^{jack-mm}_{n,\alpha}(X_{n+1}) = [g(\hat{q}^{-}_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1})), g(\hat{q}^{+}_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))]$$

Similarly to the theorem's proof, let's suppose that $Y_{n+1} \notin \hat{C}^{jack-mm}_{n,\alpha}(X_{n+1})$. This means either:

$$Y_{n+1} > g(\hat{q}^{+}_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))$$

which implies that $Y_{n+1} > g(R_j^{LOO}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))$ for at least $(1-\frac{\alpha}{2})(n+1)$ many indices $j \in \{1, ..., n\}$ since g is an increasing function regarding the first variable, or otherwise:

$$Y_{n+1} < g(\hat{q}^{-}_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1}))$$

which implies that $Y_{n+1} < g(R_j^{LOO}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1}))$ for at least $(1-\frac{\alpha}{2})(n+1)$ many indices $j \in \{1, ..., n\}$.

Here R_j^{LOO} can be positive or negative so it is added in both cases. Besides, $\frac{\alpha}{2}$ is used instead of α since to get the $(1-\alpha)$ confidence interval, we look for an upper bound on the confidence interval so that $\frac{\alpha}{2}$ of the residuals are above and a lower bound on the confidence interval so that $\frac{\alpha}{2}$ of the residuals are below.

Let's consider the first case $Y_{n+1} > g(R_j^{LOO}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))$. We have:

$$\begin{aligned}
(1-\frac{\alpha}{2})(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} > g(R_j^{LOO}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1})) > f(g(R_j^{LOO}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1})), \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{\underset{i}{\min}f(Y_{n+1}, \hat{\mu}_{-i}(X_{n+1})) = f(Y_{n+1}, \underset{i}{\max}\hat{\mu}_{-i}(X_{n+1})) > R_j^{LOO}\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{\underset{i}{\min}R_{n+1,i} > R_{j,n+1}\} = \sum_{j=1}^{n} A^{+}_{n+1,j} \qquad (34)
\end{aligned}$$

and therefore $n+1 \in S_+$, that is, point $n+1$ is a strange point with high positive residuals.

Similarly, we can conclude the same for strange point with high negative residuals.

$$\begin{aligned}
(1-\frac{\alpha}{2})(n+1) &\leq \sum_{j=1}^{n} \mathbb{1}\{Y_{n+1} < g(R_j^{LOO}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{f(Y_{n+1}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1})) < f(g(R_j^{LOO}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1})), \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1}))\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{\underset{i}{\max}f(Y_{n+1}, \hat{\mu}_{-i}(X_{n+1})) = f(Y_{n+1}, \underset{i}{\min}\hat{\mu}_{-i}(X_{n+1})) < R_j^{LOO}\} \\
&= \sum_{j=1}^{n} \mathbb{1}\{\underset{i}{\max}R_{n+1,i} < R_{j,n+1}\} = \sum_{j=1}^{n} A^{-}_{n+1,j} \qquad (35)
\end{aligned}$$

and therefore $n+1 \in S_-$, that is, point $n+1$ is a strange point with high negative residuals.

Combining with step 2, we have:

$$\begin{aligned}
\mathbb{P}\{Y_{n+1} > g(\hat{q}^+_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \max_i \hat{\mu}_{-i}(X_{n+1}))\} \leq \frac{\alpha}{2} \\
\mathbb{P}\{Y_{n+1} < g(\hat{q}^-_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \min_i \hat{\mu}_{-i}(X_{n+1}))\} \leq \frac{\alpha}{2}
\end{aligned} \tag{36}$$

Since any point cannot be simultaneously in S_+ and S_- because the boundaries are even larger than in the jackknife+ case, we finally have:

$$\begin{aligned}
\mathbb{P}\{Y_{n+1} \in \hat{C}^{jack-mm}_{n,\alpha}(X_{n+1})\} = 1 & \\
& - \mathbb{P}\{Y_{n+1} > g(\hat{q}^+_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \max_i \hat{\mu}_{-i}(X_{n+1}))\} \\
& - \mathbb{P}\{Y_{n+1} < g(\hat{q}^-_{n,\frac{\alpha}{2}}\{R_j^{LOO}\}, \min_i \hat{\mu}_{-i}(X_{n+1}))\} \\
& \geq 1 - \alpha
\end{aligned} \tag{37}$$