ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230282

Leveraging Error Patterns to Correct Prediction Intervals

Thomas Bonnier[a;*]

[a]Centrale Lille Alumni

Abstract. When assessing uncertainty in model predictions, it is key to consider potential error patterns in some regions of the feature space. In this paper, we build on quantile regression to propose a new method to produce prediction intervals in regression tasks. It estimates a conditional quantile function of the residual variable given a specific representation. The method then adjusts the regressor's prediction with an upper and lower conditional quantile prediction in order to produce an adaptive prediction interval for any new input. Further, we suggest an additional layer based on conformal prediction in order to provide coverage guarantees. Lastly, as distribution-free conditional coverage is impossible to achieve, we suggest a tree-based representation which displays patterns of undercoverage. This diagnostic tool aims to reveal which regions of the feature space are significantly less likely to have trustworthy prediction intervals. In order to prove their efficacy, our techniques are tested over various use cases and compared against four main baselines. Our methods empirically achieve the expected coverage and tend to produce shorter intervals.

1 Introduction

Models can exhibit good overall performance but they may sometimes be inclined to failure modes when error patterns surface during inference [23]. For example, a regression model could be subject to a local overestimation bias or high uncertainty in a given area of the feature space. These patterns turn out to be harmful in regression applications such as price prediction in finance or treatment effect evaluation in the medical field. When the model predicts on new data, it is thus key to produce predictive confidence intervals which are likely to contain the ground truth with a certain probability. For instance, an algorithm for real estate investors should be able to estimate an interval that would include, with a certain likelihood, the future sale price of a house based on its characteristics. The model users would also like to know whether the prediction interval tends to undercover for these types of properties. For a patient with high cholesterol level, a doctor aims to predict the effect of a treatment. He intends to obtain a prediction interval for this patient's cholesterol level in six months with a certain probability. Another requirement is to know whether the prediction interval usually covers the true response with the expected probability for this category of patients.

Objectives We consider a training dataset $\{(X_i, Y_i)\}_{i=1}^n$, indexed by a set $\mathcal{I} = \{1, ..., n\}$, with n pairs of observations drawn exchangeably (e.g. i.i.d.) from an unknown joint distribution P_{XY} over d features $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and the response $Y \in \mathcal{Y} \subseteq \mathbb{R}$. Given a new input $X_{n+1} \in \mathcal{X}$, a natural regression task is to estimate the value of the response Y_{n+1} leveraging a predictor $\hat{\mu} : \mathcal{X} \to \mathcal{Y}$, where (X_{n+1}, Y_{n+1}) is also drawn exchangeably from P_{XY}. The regression function $\hat{\mu}$ estimates the conditional mean of the response. Beyond the point-wise prediction, the uncertainty can be assessed with a prediction interval $\hat{C}(X_{n+1}) \subseteq \mathcal{Y}$. We say that $\hat{C}$ satisfies *distribution-free marginal coverage* if it includes the true response Y_{n+1} with probability:

$$\mathbb{P}\{Y_{n+1} \in \hat{C}(X_{n+1})\} \geq 1 - \alpha, \text{for all } P_{XY}. \quad (1)$$

$1-\alpha$ denotes the target coverage level (e.g. 90%) and the probability is over $\{(X_i, Y_i)\}_{i=1}^{n+1}$. Further, we say that $\hat{C}$ satisfies *distribution-free conditional coverage* if, for all P_{XY} and almost all x:

$$\mathbb{P}\{Y_{n+1} \in \hat{C}(X_{n+1}) | X_{n+1} = x\} \geq 1 - \alpha. \quad (2)$$

In this paper, we seek to construct prediction intervals with empirical coverage level of $1 - \alpha$. An additional requirement is that those intervals should be short and adaptive, based on local patterns detected in the residuals $R_i = Y_i - \hat{\mu}(X_i)$. Lastly, as assessing conditional coverage is critical for some applications, we would like to understand the limitations of our method by displaying the regions of the feature space where the desired coverage level does not hold.

Novelty of the proposed approaches To achieve those objectives, we build on quantile regression [11], conformal prediction [34, 17, 14], and conformalized quantile regression [21]. Our method is illustrated in Figure 1 and is a sequence of three techniques. A regression model $\hat{\mu}$ is fitted to the training dataset. The first technique is called *quantile regression+ (QR+)* as it aims to adjust the regressor's prediction with the estimates of an upper and lower conditional quantile of the residual variable given $(X, \hat{\mu}(X))$. We thus fit two conditional quantile regressors on a separate residual dataset with the targeted bounds ($\alpha/2$ and $(1 - \alpha/2)$ quantiles). The residual dataset can be thought of as a *multidimensional residual plot:* $\{((X_i, \hat{\mu}(X_i)), R_i)\}$. Given any new input, QR+ then generates a prediction interval by adding each of the conditional quantile predictions to the regressor's prediction. The *conformalized quantile regression+ (CQR+)* complements QR+ with a layer based on conformal prediction in order to accurately adapt the width of the prediction interval based on the desired coverage level. Lastly, the *undercoverage tree analysis (UTA)* is a tree-based representation which displays the paths to regional undercoverage. This diagnostic tool aims to provide practitioners with miscoverage warnings for certain specific regions of the feature space.

* Corresponding Author. Email: thomas.bonnier@centraliens-lille.org

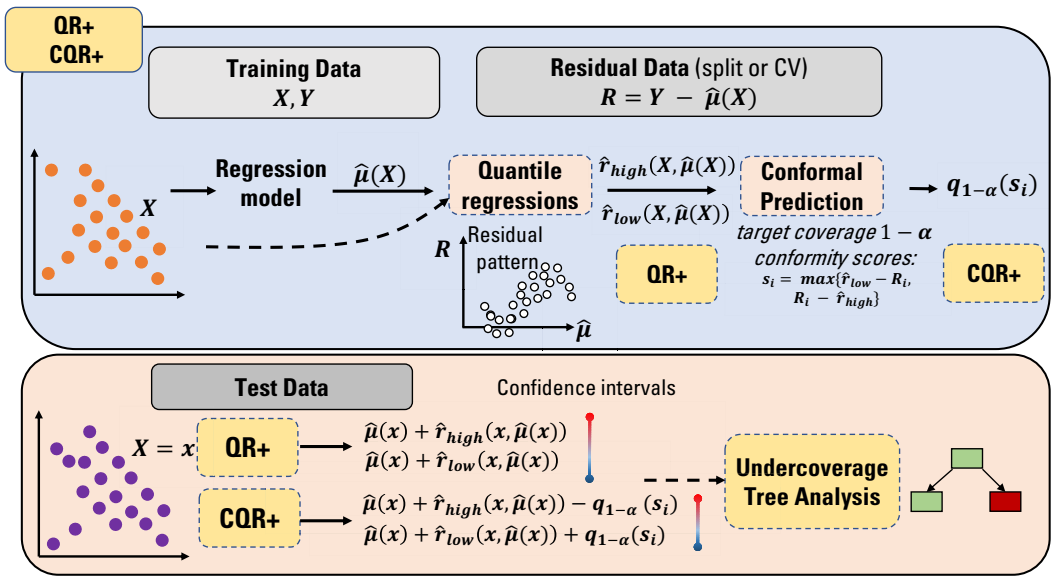

Figure 1. Illustration of our method. *Top:* We first fit the regression function $\hat{\mu}$. To construct QR+, we fit an upper and lower conditional quantile regression model to a residual dataset $\{((X_i, \hat{\mu}(X_i)), R_i)\}$. CQR+ completes QR+ with a conformal prediction layer and produces a quantile of the conformity scores. Our method can work with a split or cross-validation approach. *Bottom:* Given a new input, QR+ and CQR+ output prediction intervals by adding each component previously estimated. The undercoverage tree analysis reveals regions of the feature space where the prediction intervals tend to undercover.

Our techniques thus learn from the original regressor's mistakes through the residual dataset. They produce prediction intervals based on the assessment of the correction to be made. The objective is thus to detect error patterns in the behavior of the regressor in order to predict the lower and upper bounds in an adaptive fashion. The main difference between our techniques and (conformalized) quantile regression is that the former learn from the residuals computed on a separate dataset instead of the original data. For instance, the quantile regression estimates conditional quantiles of Y while QR+ estimates conditional quantiles of the residual variable R. The conformalized quantile regression leverages a conformity score based on the response and estimated conditional quantiles of Y whereas CQR+ employs a score based on the true residuals and estimated conditional quantiles of R. The advantage of working with residuals is that they can reveal the original regressor's failure modes (i.e. bias, heteroscedasticity) that our methods can leverage.

Notations The quantiles $q^+_{\alpha,|\mathcal{I}|}\{v_i\}$ and $q^-_{\alpha,|\mathcal{I}|}\{v_i\}$ respectively denote the $1-\alpha$ and α quantiles of values $\{v_i : i \in \mathcal{I}\}$, where $\mathcal{I}$ denotes a set of $|\mathcal{I}|$ indices. $q^+_{\alpha,|\mathcal{I}|}\{v_i\}$ is thus the $\lceil(1-\alpha)(|\mathcal{I}|+1)\rceil$-th smallest value of $\{v_i : i \in \mathcal{I}\}$, and $q^-_{\alpha,|\mathcal{I}|}\{v_i\} = -q^+_{\alpha,|\mathcal{I}|}\{-v_i\}$. Let $\mathcal{R}$ denote any regression algorithm that takes in training data indexed by $\mathcal{I}$ in order to output a regression model fitted on that data: $\hat{\mu} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}\})$. Similarly, let $\mathcal{Q}$ be any quantile regression algorithm. Lastly, $\hat{\mu}^{-\mathcal{I}_k} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}\backslash\mathcal{I}_k\})$ is the model fitted on the training dataset after removing the subset indexed by $\mathcal{I}_k$.

Background Some conformal prediction methods require splitting the initial training dataset into disjoint subsets, i.e. training data (for $\hat{\mu}$) indexed by a set of indices $\mathcal{I}_1$ and calibration data (for conformity scores) indexed by $\mathcal{I}_2$. For instance, the *split* or *inductive conformal prediction* [17, 16] produces confidence intervals around the regression function's predictions, adjusted with a quantile of the distribution of the absolute residuals computed on the calibration set indexed by $\mathcal{I}_2$: $[\hat{\mu}(X_{n+1}) \pm q^+_{\alpha,|\mathcal{I}_2|}\{|Y_i - \hat{\mu}(X_i)|\}]$. As demonstrated in [16, 32], (1) is satisfied.

Unlike the previous technique, methods based on cross-validation (leave-one-out can be considered as a special case) do not require a separate calibration dataset. For instance, *CV+ for K-fold cross-validation (CV+)* [2] considers the variability of the regression models. The training dataset is split into K disjoint subsets indexed by $\mathcal{I}_1, ..., \mathcal{I}_K$. K regression models $\hat{\mu}^{-\mathcal{I}_k}$ are fitted to the training data after removing the k-th subset indexed by $\mathcal{I}_k$. The absolute residuals are computed with the adequate regression model for each data point i of $\{(X_i, Y_i)\}_{i=1}^n$: $R_i^{CV} = |Y_i - \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)|$ where $i \in \mathcal{I}_{k(i)}$, with $k(i) \in \{1, ..., K\}$. Lastly, it predicts the following interval given a new input X_{n+1}: $[q^-_{\alpha,|\mathcal{I}|}\{\hat{\mu}^{-\mathcal{I}_{k(i)}}(X_{n+1}) - R_i^{CV}\}, q^+_{\alpha,|\mathcal{I}|}\{\hat{\mu}^{-\mathcal{I}_{k(i)}}(X_{n+1}) + R_i^{CV}\}]$. It guarantees a theoretical coverage level of at least $1-2\alpha$. The *jackknife+* method [2] is a specific case of CV+ with a leave-one-out approach. The *jackknife+-after-bootstrap method (J+aB)* can be viewed as an alternative based on bootstraps. It also attains a $1-2\alpha$ theoretical coverage guarantee [10]. This method creates B training datasets indexed by $\mathcal{I}_1, ..., \mathcal{I}_B$ by bootstrapping from the available training data. B regression functions $\hat{\mu}_b$ are then fitted on these bootstraps. For each data point i, there is an aggregation agg (e.g. mean) of the predictions of the $\hat{\mu}_b$'s whose training dataset indexed by $\mathcal{I}_b$ did not contain this point: $\hat{\mu}_{agg\backslash i} = agg(\{\hat{\mu}_b : b = 1, ..., B, \mathcal{I}_b \not\ni i\})$. The absolute residuals can then be computed for the i-th data point ($i \in \mathcal{I}$) with the related aggregation function: $R_i^{J+aB} = |Y_i - \hat{\mu}_{agg\backslash i}(X_i)|$. Lastly, the prediction interval is $\hat{C}(X_{n+1}) = [q^-_{\alpha,|\mathcal{I}|}\{\hat{\mu}_{agg\backslash i}(X_{n+1}) - R_i^{J+aB}\}, q^+_{\alpha,|\mathcal{I}|}\{\hat{\mu}_{agg\backslash i}(X_{n+1}) + R_i^{J+aB}\}]$.

Another alternative to produce confidence intervals is the *conditional quantile regression (QR)* [11]. Its objective is to estimate the α-th conditional quantile function of Y given $X = x$, defined as $q_\alpha(x) = \inf\{y : F_{Y|X}(y|X = x) \geq \alpha\}$. $F_{Y|X}(y|X = x)$ is the conditional distribution function of Y given $X = x$: $\mathbb{P}\{Y \leq y|X =$

$x\}$. This estimation is performed by minimizing the quantile loss (or pinball loss [24]) on the training data:

$$\mathcal{L}_\alpha(\hat{q}_\alpha(x), y) = (y - \hat{q}_\alpha(x))\alpha \mathbb{1}[y > \hat{q}_\alpha(x)] + (\hat{q}_\alpha(x) - y)(1-\alpha)\mathbb{1}[y \leq \hat{q}_\alpha(x)].$$

This approach can be leveraged to produce the prediction interval $[\hat{q}_{\alpha/2}(X_{n+1}), \hat{q}_{1-\alpha/2}(X_{n+1})]$ for a target coverage level of $1-\alpha$, with the lower and upper bounds being the estimates of the $\alpha/2$-th and $(1-\alpha/2)$-th conditional quantiles, respectively. However, there is no theoretical guarantee to satisfy (1) [21]. In order to fulfill (1), Romano et al. [21] suggest combining the strengths of conformal prediction and quantile regression. This method, called *conformalized quantile regression (CQR)*, produces valid and adaptive predictive intervals, with length varying according to heteroscedasticity in the data. The authors describe the split and symmetric version of CQR. Given a new input X_{n+1}, it constructs the prediction interval by leveraging the estimates of conditional quantiles and a quantile of conformity scores s_i: $[\hat{q}_{\alpha/2}(X_{n+1}) - q^+_{\alpha,|\mathcal{I}_2|}\{s_i\}, \hat{q}_{1-\alpha/2}(X_{n+1}) + q^+_{\alpha,|\mathcal{I}_2|}\{s_i\})]$. $\hat{q}_{\alpha/2}$ and $\hat{q}_{1-\alpha/2}$ have been fitted on a training dataset and the scores s_i have been computed on a calibration dataset indexed by $\mathcal{I}_2$: $s_i = \max\{\hat{q}_{\alpha/2}(X_i) - Y_i, Y_i - \hat{q}_{1-\alpha/2}(X_i)\}$.

Lastly, it is known that distribution-free conditional coverage is impossible to achieve for any finite-width interval [8]. The authors thus propose an algorithm with approximate conditional coverage property which is valid for sets of similar instances (e.g. similar patients in the medical example). Regional uncertainty can also be explained by leveraging tree representations [3].

2 Methods

In this section, we detail our methods with *split* and *cross-validation* (CV) versions for both QR+ and CQR+. They are respectively called QR+ (SPLIT), CQR+ (SPLIT), QR+ (CV), and CQR+ (CV). The split version requires splitting the training dataset into subsets while the CV option requires fitting the regression model multiple times.

For the sake of clarity, $\hat{q}_\alpha$ denotes a conditional quantile model fitted on the original training data $\{(X_i, Y_i)\}$, whereas $\hat{r}_\alpha$ indicates a conditional quantile model fitted on the residual dataset $\{((X_i, \hat{\mu}(X_i)), R_i)\}$.

2.1 QR+ (split)

Our objective is to produce prediction intervals with characteristics stated in Section 1. By definition, $Y_{n+1} = \hat{\mu}(X_{n+1}) + R_{n+1}$, where R_{n+1} is the (net) residual error for the input X_{n+1}. Instead of directly estimating a conditional quantile of Y_{n+1} given $X_{n+1} = x$, we consider estimating a conditional quantile of R_{n+1}. To achieve that, QR+ seeks to detect any pattern in the residuals given a specific representation $(X, \hat{\mu}(X))$. The residual dataset $\{((X_i, \hat{\mu}(X_i)), R_i)\}$ could be thought of as a multidimensional residual plot. The residual plot, showing the residual values against the fitted values or against the covariate values, is a diagnostic graph employed to check whether the variance of residuals is constant in linear regression models [1]. It can also be employed to detect other patterns such as a local bias. As $\hat{\mu}(X_{n+1}) + \hat{R}_{n+1}$ can be considered as an attempt to correct the regressor's initial prediction, estimating upper and lower bounds could be viewed as the predictive confidence in the applied correction: $\hat{\mu}(X_{n+1}) + \hat{r}_{\alpha/2}(X_{n+1}, \hat{\mu}(X_{n+1}))$ for the lower bound and $\hat{\mu}(X_{n+1}) + \hat{r}_{1-\alpha/2}(X_{n+1}, \hat{\mu}(X_{n+1}))$ for the upper bound, where $\{\hat{r}_{\alpha/2}, \hat{r}_{1-\alpha/2}\} = \mathcal{Q}(\{((X_i, \hat{\mu}(X_i)), R_i)\})$. It is worth noting that our method does not make any assumption on the properties of the residuals. The residual structure could be biased/unbiased or hetero/homoscedastic.

QR+ thus requires a residual dataset where each residual R_i is computed with a regression model fitted on the training dataset that does not contain the i-th data point. We describe here the different steps of the split version:

- We first split the training dataset $\{(X_i, Y_i)\}_{i=1}^n$ into two disjoint subsets, indexed by $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively.
- We fit the regression function on the first set: $\hat{\mu} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}_1\})$.
- We compute the (net) residuals on $\{(X_i, Y_i) : i \in \mathcal{I}_2\}$: $R_i = Y_i - \hat{\mu}(X_i)$.
- We fit two conditional quantile regression functions $\hat{r}$ on the residual dataset: $\{\hat{r}_{\alpha/2}, \hat{r}_{1-\alpha/2}\} = \mathcal{Q}(\{((X_i, \hat{\mu}(X_i)), R_i) : i \in \mathcal{I}_2\})$.
- Given a new input $X_{n+1} = x$, QR+ constructs $\hat{C}(x)$:

$$[\hat{\mu}(x) + \hat{r}_{\alpha/2}(x, \hat{\mu}(x)), \hat{\mu}(x) + \hat{r}_{1-\alpha/2}(x, \hat{\mu}(x))].$$

Similarly to QR, there is no theoretical guarantee that (1) is satisfied.

2.2 CQR+ (split)

In order to obtain theoretical coverage guarantees, we conformalize QR+. We adapt the conformity scores proposed in [21] to our method based on residuals. The split version of CQR+ is carried out as follows:

- We first split the training dataset $\{(X_i, Y_i)\}_{i=1}^n$ into three disjoint subsets, indexed by $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$, respectively.
- $\hat{\mu}$ is fitted on the first dataset: $\hat{\mu} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}_1\})$.
- We compute the (net) residuals on $\{(X_i, Y_i) : i \in \mathcal{I}_2\}$: $R_i = Y_i - \hat{\mu}(X_i)$.
- Two conditional quantile regression functions are fitted on the residual dataset: $\{\hat{r}_{\alpha/2}, \hat{r}_{1-\alpha/2}\} = \mathcal{Q}(\{((X_i, \hat{\mu}(X_i)), R_i) : i \in \mathcal{I}_2\})$.
- The conformity scores are computed on $\{(X_i, Y_i) : i \in \mathcal{I}_3\}$: $s_i = \max\{\hat{r}_{\alpha/2}(X_i, \hat{\mu}(X_i)) - R_i, R_i - \hat{r}_{1-\alpha/2}(X_i, \hat{\mu}(X_i))\}$
- We compute the quantile of the conformity scores $q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}$, with $i \in \mathcal{I}_3$.
- Given a new input $X_{n+1} = x$, the method constructs $\hat{C}(x)$:

$$[\hat{\mu}(x) + \hat{r}_{\alpha/2}(x, \hat{\mu}(x)) - q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}, \hat{\mu}(x) + \hat{r}_{1-\alpha/2}(x, \hat{\mu}(x)) + q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}].$$

Theorem 1 *If $(X_i, Y_i), i = 1, ..., n+1$ are exchangeable, then the output $\hat{C}(X_{n+1})$ of the CQR+ (SPLIT) method satisfies (1).*

Proof. The proof follows the main ideas of the split conformal prediction guarantee [16, 32, 13]. Let $\hat{C}_r(X_{n+1})$ denote the prediction interval for the residual R_{n+1}, i.e. $[\hat{r}_{\alpha/2}(X_{n+1}, \hat{\mu}(X_{n+1})) - q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}, \hat{r}_{1-\alpha/2}(X_{n+1}, \hat{\mu}(X_{n+1})) + q^+_{\alpha,|\mathcal{I}_3|}\{s_i\})]$. We note that $\{R_{n+1} \in \hat{C}_r(X_{n+1})\} = \{s_{n+1} \leq q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}\}$. Conditioning on the training sets indexed by $\mathcal{I}_1 \cup \mathcal{I}_2$, if (X_i, Y_i) are exchangeable, so are the conformity scores s_i for $i \in \mathcal{I}_3$ and $i = n+1$. We have: $\mathbb{P}\{s_{n+1} \leq q^+_{\alpha,|\mathcal{I}_3|}\{s_i\} | (X_k, Y_k) : k \in \mathcal{I}_1 \cup \mathcal{I}_2\} = \lceil (|\mathcal{I}_3|+1)(1-\alpha) \rceil / (|\mathcal{I}_3|+1) \geq 1-\alpha$, by definition of quantile $q^+_{\alpha,|\mathcal{I}_3|}\{s_i\}$. As $R_{n+1} = Y_{n+1} - \hat{\mu}(X_n + 1)$, we have $\mathbb{P}\{(Y_{n+1} - \hat{\mu}(X_n+1)) \in \hat{C}_r(X_{n+1}) | (X_k, Y_k) : k \in \mathcal{I}_1 \cup \mathcal{I}_2\} \geq 1-\alpha$, and thus $\mathbb{P}\{Y_{n+1} \in \hat{C}(X_{n+1}) | (X_k, Y_k) : k \in \mathcal{I}_1 \cup \mathcal{I}_2\} \geq 1-\alpha$. We conclude by taking the expectation over the training sets.

Algorithm 1 QR+ (CV)

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$ indexed by $\mathcal{I}$, number of folds K, target coverage $1-\alpha$
Output: Prediction interval $\hat{C}$

Split data into K disjoint subsets indexed by $\mathcal{I}_1, ..., \mathcal{I}_K$.
for $k = 1, ..., K$ **do**
  Fit $\hat{\mu}^{-\mathcal{I}_k} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I} \backslash \mathcal{I}_k\})$.
end for
for $i = 1, ..., n$ **do**
  Compute the residual $R_i = Y_i - \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)$.
end for
Fit conditional quantile regression models $\{\hat{r}_{\alpha/2}, \hat{r}_{1-\alpha/2}\} = \mathcal{Q}(\{((X_i, \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)), R_i) : i \in \mathcal{I}\})$.
Fit on full data $\hat{\mu} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}\})$.
Compute prediction interval for any $X_{n+1} = x$:
$[\hat{\mu}(x) + \hat{r}_{\alpha/2}(x, \hat{\mu}(x)), \hat{\mu}(x) + \hat{r}_{1-\alpha/2}(x, \hat{\mu}(x))]$.

2.3 QR+ (CV)

The computational cost of the split version is relatively low as it only requires fitting three functions ($\hat{\mu}$, $\hat{r}_{\alpha/2}$, and $\hat{r}_{1-\alpha/2}$). However, these versions require splitting the initial training dataset. Fitting $\hat{\mu}$ and $\hat{r}$ on small subsets can produce poor models of the data and thus wide prediction intervals. Further, a small calibration dataset (e.g. subset indexed by $\mathcal{I}_3$ for CQR+) can produce unreliable conformity scores with high variability [33]. On the other hand, QR+ (CV) is based on cross-validation and does not require splitting the training dataset. According to Kohavi [12], if the regressor remains stable under the perturbations created by removing each fold, the cross-validation estimate of the generalization error will remain unbiased. Better stability should be observed with a large number (K) of subsets. In our experiments, we followed Kohavi's suggestion to use ten folds as a reasonable trade-off between stability and computational cost.

QR+ (CV) is described in Algorithm 1. $\mathcal{I}_{k(i)}$ contains the i-th data point, with $k(i) \in \{1, ..., K\}$. Therefore, in order to calculate each residual R_i, the prediction $\hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)$ is computed with the regression model fitted on the training set that does not contain the i-th data point. To fit the two conditional quantile regression models, the covariate $\hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)$ is computed with the regression function fitted on the training set that does not contain the data point i.

2.4 CQR+ (CV)

This version of CQR+ is based on cross-validation and thus does not require splitting the training dataset either. CQR+ (CV) is described in Algorithm 2. As mentioned previously for QR+ (CV), $\mathcal{I}_{k(i)}$ contains the i-th data point, with $k(i) \in \{1, ..., K\}$. Therefore, in order to compute each conformity score s_i, we use the functions ($\hat{\mu}^{-\mathcal{I}_{k(i)}}$, $\hat{r}_{\alpha/2}^{-\mathcal{I}_{k(i)}}$, $\hat{r}_{1-\alpha/2}^{-\mathcal{I}_{k(i)}}$) fitted on the training set that does not contain the data point i.

Computational cost If we consider QR+ (CV), the model training cost is $K+3$ because we fit K $\hat{\mu}^{-\mathcal{I}_k}$, two conditional quantile regression models, and $\hat{\mu}$. In terms of prediction, the model evaluation cost is 3 in order to compute $\hat{\mu}(.)$, $\hat{r}_{\alpha/2}(.)$, and $\hat{r}_{1-\alpha/2}(.)$. With regard to CQR+ (CV), the training cost is $3K+3$ in order to fit K $\hat{\mu}^{-\mathcal{I}_k}$, $2K$ conditional quantile regression models, then $\hat{\mu}$, $\hat{r}_{\alpha/2}$, and $\hat{r}_{1-\alpha/2}$. In that case, the model evaluation cost is 3 as well.

Algorithm 2 CQR+ (CV)

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$ indexed by $\mathcal{I}$, number of folds K, target coverage $1-\alpha$
Output: Prediction interval $\hat{C}$

Split Data into K disjoint subsets indexed by $\mathcal{I}_1, ..., \mathcal{I}_K$.
for $k = 1, ..., K$ **do**
  Fit $\hat{\mu}^{-\mathcal{I}_k} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I} \backslash \mathcal{I}_k\})$.
end for
for $i = 1, ..., n$ **do**
  Compute the residual $R_i = Y_i - \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)$.
end for
for $k = 1, ..., K$ **do**
  Fit cond. quantile regressors $\{\hat{r}_{\alpha/2}^{-\mathcal{I}_k}, \hat{r}_{1-\alpha/2}^{-\mathcal{I}_k}\} = \mathcal{Q}(\{((X_i, \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)), R_i) : i \in \mathcal{I} \backslash \mathcal{I}_k\})$.
end for
for $i = 1, ..., n$ **do**
  Compute conformity scores $s_i = \max\{\hat{r}_{\alpha/2}^{-\mathcal{I}_{k(i)}}(X_i, \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)) - R_i, R_i - \hat{r}_{1-\alpha/2}^{-\mathcal{I}_{k(i)}}(X_i, \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i))\}$.
end for
Compute the quantile $q^+_{\alpha,|\mathcal{I}|}\{s_i\}$.
Fit on full data $\{\hat{r}_{\alpha/2}, \hat{r}_{1-\alpha/2}\} = \mathcal{Q}(\{((X_i, \hat{\mu}^{-\mathcal{I}_{k(i)}}(X_i)), R_i) : i \in \mathcal{I}\})$.
Fit on full data $\hat{\mu} = \mathcal{R}(\{(X_i, Y_i) : i \in \mathcal{I}\})$.
Compute prediction interval for any $X_{n+1} = x$:
$[\hat{\mu}(x) + \hat{r}_{\alpha/2}(x, \hat{\mu}(x)) - q^+_{\alpha,|\mathcal{I}|}\{s_i\}, \hat{\mu}(x) + \hat{r}_{1-\alpha/2}(x, \hat{\mu}(x)) + q^+_{\alpha,|\mathcal{I}|}\{s_i\}]$.

Stability CQR+ (CV) does not have any theoretical coverage guarantees. In some specific settings (e.g. $n \approx d$), $\hat{\mu}$ could be unstable. In such contexts, the *jackknife* method, for instance, may have significant undercoverage compared to *jackknife+* [2]. We thus wonder whether the intervals produced by CQR+ (CV) may undercover in these conditions. Indeed, CQR+ (CV) computes the conformity scores with $\hat{\mu}^{-\mathcal{I}_{k(i)}}$, $\hat{r}_{\alpha/2}^{-\mathcal{I}_{k(i)}}$, and $\hat{r}_{1-\alpha/2}^{-\mathcal{I}_{k(i)}}$ whereas it uses $\hat{\mu}$, $\hat{r}_{\alpha/2}$, and $\hat{r}_{1-\alpha/2}$ to construct the prediction intervals. In a context of instability, we would recommend producing the prediction intervals with the same $\hat{\mu}^{-\mathcal{I}_k}$ and $\hat{r}^{-\mathcal{I}_k}$:

$$[q^-_{\alpha,|\mathcal{I}|}\{\hat{\mu}^{-\mathcal{I}_{k(i)}}(x) + \hat{r}_{\alpha/2}^{-\mathcal{I}_{k(i)}}(x, \hat{\mu}^{-\mathcal{I}_{k(i)}}(x)) - s_i\},$$
$$q^+_{\alpha,|\mathcal{I}|}\{\hat{\mu}^{-\mathcal{I}_{k(i)}}(x) + \hat{r}_{1-\alpha/2}^{-\mathcal{I}_{k(i)}}(x, \hat{\mu}^{-\mathcal{I}_{k(i)}}(x)) + s_i\}].$$

2.5 The undercoverage tree analysis

As distribution-free conditional coverage is impossible to achieve for any finite-width interval [8], we propose the undercoverage tree analysis (UTA) in order to discover potential patterns of undercoverage. This tool aims to reveal regions of the feature space where the prediction intervals are more likely to undercover. UTA has to be used on a dataset (indexed by $\mathcal{I}_{uta}$) where the response values are available. UTA should be considered as a diagnostic tool giving insights to model users. The latter should be informed about the model caveats, i.e. whether the prediction intervals are really trustworthy or not.

Let $\mathcal{T}$ denote a classification tree algorithm [4] that takes in training data indexed by $\mathcal{I}_{uta}$ in order to output a model fitted on that data: $\hat{t} = \mathcal{T}(\{(X_j, U_j) : j \in \mathcal{I}_{uta}\})$. The label $U_j \in \{0, 1, 2\}$

Table 1. Marginal coverage (%) computed on the test dataset and averaged over ten training-test splits. The results are displayed by dataset, by base algorithm (*lgbm*, *linear*), and for each method. The last two lines represent a simple mean across datasets. Values in parenthesis are standard deviations. All methods empirically achieve roughly 90% coverage.

Dataset	Model	QR	QR+ (ours)	CQR	CQR+ (ours)	CV+	J+aB
house	lgbm	90.10 (1.95)	90.75 (1.94)	90.38 (1.60)	89.76 (1.81)	91.61 (1.91)	90.65 (2.41)
	linear	91.30 (1.35)	90.96 (2.12)	90.96 (1.33)	90.14 (1.60)	90.58 (1.34)	90.17 (0.94)
bike	lgbm	90.64 (0.91)	90.78 (0.62)	90.29 (0.66)	90.76 (0.40)	91.59 (0.79)	90.54 (0.71)
	linear	90.72 (1.36)	90.33 (1.31)	90.32 (2.03)	89.94 (1.00)	89.79 (0.95)	89.87 (0.87)
community	lgbm	90.48 (3.34)	90.83 (1.77)	89.77 (1.36)	90.40 (1.56)	90.85 (1.99)	90.13 (1.96)
	linear	97.37 (4.54)	89.82 (3.82)	91.03 (1.79)	90.15 (1.72)	90.30 (1.43)	90.28 (1.56)
concrete	lgbm	88.81 (2.77)	93.28 (2.53)	91.00 (2.34)	92.99 (2.16)	93.98 (1.91)	91.94 (2.62)
	linear	91.04 (2.78)	91.79 (2.85)	90.05 (1.95)	90.30 (2.26)	91.09 (1.94)	91.04 (1.86)
fb1	lgbm	91.10 (1.17)	90.16 (0.46)	93.52 (1.37)	90.05 (0.34)	92.02 (0.65)	90.74 (0.38)
	linear	92.59 (0.96)	89.87 (1.01)	94.55 (0.80)	89.59 (1.02)	89.85 (0.85)	89.80 (0.90)
protein	lgbm	90.02 (0.65)	90.53 (0.31)	90.02 (0.15)	90.44 (0.29)	90.61 (0.62)	89.99 (0.45)
	linear	90.71 (1.17)	90.79 (0.93)	89.78 (0.97)	90.20 (0.81)	89.88 (0.79)	89.89 (0.82)
simple mean	lgbm	90.19	91.06	90.83	90.73	91.78	90.66
	linear	92.29	90.59	91.11	90.05	90.25	90.18

is defined as $U_j = 2.1\!\!1(Y_j > UB_j) + 1\!\!1(Y_j < LB_j)$. LB_j and UB_j are respectively the lower and upper bounds of the prediction interval generated by CQR+ (CV) for input $X_j \in \mathcal{I}_{uta}$, i.e. $LB_j = \hat{\mu}(X_j) + \hat{r}_{\alpha/2}(X_j, \hat{\mu}(X_j)) - q^+_{\alpha,|\mathcal{I}|}\{s_i\}$, and $UB_j = \hat{\mu}(X_j) + \hat{r}_{1-\alpha/2}(X_j, \hat{\mu}(X_j)) + q^+_{\alpha,|\mathcal{I}|}\{s_i\}$, with $i \in \mathcal{I}$. The labels 1 and 2 identify when the prediction intervals do not contain the response, with CQR+ overestimating the response ($Y_j < LB_j$) or underestimating the response ($Y_j > UB_j$), respectively. Indeed, it can be useful to characterize the miscoverage depending on the use case at hand.

We are interested in displaying the regions of the feature space where the prediction intervals tend to undercover. We know that the leaf nodes in a tree can be defined by not necessarily closed hyperrectangles [19], which form a partition $\bigcup_{k\geq 1} H_{\hat{t}}(l_k)$ of the feature space. They are defined as $H_{\hat{t}}(l_k) = \{x \in \mathcal{X} | cst_{\hat{t}}(l_k) \models x\}$ for all k, where $\hat{t}$ is the binary decision tree (two children by internal node), l_k denotes the k-th leaf node of $\hat{t}$, and $cst_{\hat{t}}(l)$ are the constraints that fulfill the split conditions in the tree path from the root to leaf l. Given a target coverage $1-\alpha$ and tolerance δ, we would like to identify the leaf nodes with miscoverage level $\alpha_k = \mathbb{P}\{Y_j \notin \hat{C}(X_j) | X_j \in H_{\hat{t}}(l_k)\} > \alpha$ (i.e. undercoverage) and with proportion $P_X(H_{\hat{t}}(l_k)) \geq \delta$. We aim to display the leaf nodes' related size, miscoverage rate (proportions of labels 1 and 2), and constraints. By plotting the tree structure with a minimum number of samples required to be in a leaf node ($\delta \times |\mathcal{I}_{uta}|$), we can show each leaf l with its size (number of samples in the hyperrectangle), miscoverage rate (proportion of samples with label $U_j \neq 0$ in the hyperrectangle), and the combination of constraints $cst_{\hat{t}}(l)$ that *explain* that leaf node.

UTA is thus a binary tree structure which identifies the patterns that lead to regional undercoverage and explain them in terms of feature values. When used in high-dimensional settings, it could be useful to display untrustworthy regions through a combination of a few constraints. Based on UTA, practitioners could then decide to reduce the scope of use of the method (e.g. leaves with high undercoverage). If CQR+ predictions are directly communicated to non-expert users, human-in-the loop [7] could help to reduce the risks. In that case, experts would first check the prediction intervals for new inputs that would fall into the risky leaves. At least, users should be informed of the limitations related to those regions of the feature space. As a monitoring tool, UTA can thus be updated on new data as soon as the ground truth becomes available. As a last remark, UTA could work with any other method producing prediction intervals with defined target coverage (e.g. split conformal prediction).

3 Experiments

3.1 Settings

Datasets We empirically test the relevance of our methods on six regression datasets. The response in Ames Housing Dataset (*house*) is the sale price of residential properties [5, 6]. We use the 1,460 instances for which the outcome is available. The response in Seoul Bike Sharing Demand Dataset (*bike*) is the count of public bikes rented every hour in Seoul [31, 30, 29]. The dependent variable in the Communities and Crime Data Set (*community*) [20, 26] is the crime rate. The regression task proposed by the Concrete Compressive Strength Dataset (*concrete*) is to predict this strength value [35, 25]. The task corresponding to the Facebook Comment Volume variant one dataset (*fb1*) is to predict the number of comments that a post will receive [22, 28]. Lastly, the response in the Physicochemical Properties of Protein Tertiary Structure Dataset (*protein*) is the size of the residue for a protein [27].

Models and training settings The target miscoverage rate α is set to 0.1 for a target coverage of 90%. For each use case, the features are standardized to have zero mean and unit variance. We also rescale the response by dividing it by its mean absolute value. Each use case is run over ten different training-test splits. 80%/20% of the instances are used for training/testing, respectively.

We experiment with two types of base algorithms to fit $\hat{\mu}$, $\hat{q}$, or $\hat{r}$. First, we use LightGBM (*lgbm*) [9] implemented in *lightgbm* Python package. In that case, the regression model $\hat{\mu}$ is produced by optimizing the L2 loss, while the conditional quantile regression functions are fitted by optimizing the pinball loss. Secondly, we employ

Table 2. Mean interval width computed on the test dataset and averaged over ten training-test splits. The results are displayed by dataset, by base algorithm (*lgbm*, *linear*), and for each method. The last two lines represent a simple mean across datasets. Values in parenthesis are standard deviations. CQR+ leads to the shortest intervals on average while QR+ is very close. Across the various datasets and models, QR+ and CQR+ outperform QR and CQR, respectively.

Dataset	Model	QR	QR+ (ours)	CQR	CQR+ (ours)	CV+	J+aB
house	lgbm	0.68 (0.07)	0.42 (0.03)	0.53 (0.08)	0.40 (0.01)	0.41(0.02)	**0.39** (0.01)
	linear	0.73 (0.20)	0.45 (0.08)	0.65 (0.18)	**0.42** (0.05)	0.43 (0.05)	0.43 (0.04)
bike	lgbm	1.16 (0.13)	0.67 (0.04)	0.79 (0.05)	**0.66** (0.04)	0.78 (0.03)	0.77 (0.04)
	linear	2.58 (0.45)	2.23 (0.24)	2.49 (0.44)	**2.18** (0.24)	2.19 (0.19)	**2.18** (0.18)
community	lgbm	2.97 (1.06)	1.87 (0.05)	**1.63** (0.06)	1.84 (0.04)	1.84 (0.03)	1.80 (0.03)
	linear	3.77 (0.82)	1.97 (0.16)	2.17 (0.33)	**1.91** (0.17)	2.06 (0.07)	2.05 (0.06)
concrete	lgbm	0.97 (0.13)	0.41 (0.03)	0.50 (0.04)	**0.40** (0.02)	**0.40** (0.01)	**0.40** (0.01)
	linear	1.30 (0.28)	1.06 (0.11)	1.23 (0.26)	**1.00** (0.09)	1.02 (0.05)	1.02 (0.04)
fb1	lgbm	1.85 (0.08)	1.85 (0.16)	2.07 (0.08)	**1.84** (0.10)	2.17 (0.10)	1.95 (0.07)
	linear	**1.63** (0.20)	2.27 (0.54)	2.21 (0.42)	2.13 (0.47)	2.15 (0.20)	2.13 (0.22)
protein	lgbm	1.78 (0.04)	1.40 (0.10)	1.60 (0.09)	**1.33** (0.07)	1.71 (0.06)	1.69 (0.07)
	linear	2.23 (0.08)	2.22 (0.08)	2.20 (0.06)	**2.19** (0.05)	2.57 (0.18)	2.55 (0.18)
simple mean	lgbm	1.57	1.10	1.19	**1.08**	1.22	1.16
	linear	2.04	1.70	1.82	**1.64**	1.73	1.73

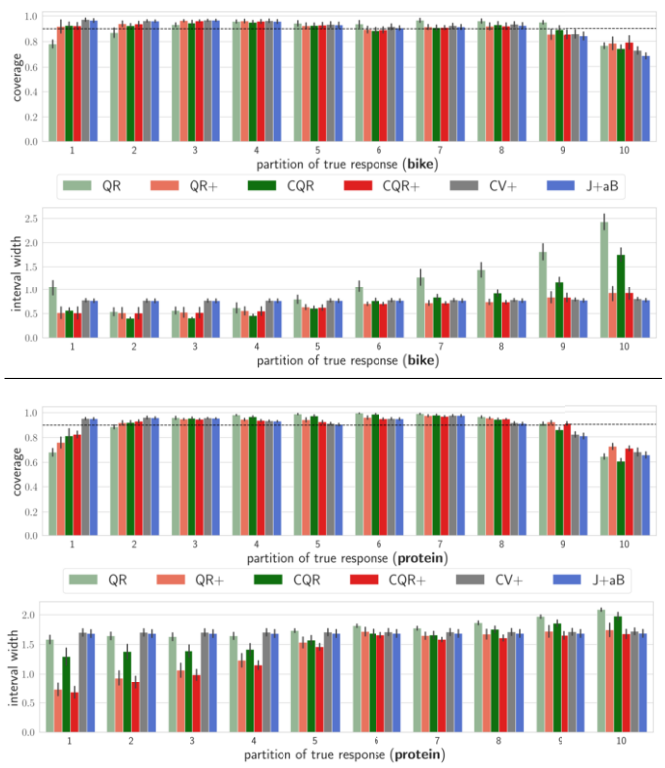

Figure 2. Conditional coverage and interval width for *bike* (*top*) and *protein* (*bottom*) use cases with *lgbm* base regression algorithm and for a quantile-based partition of the true response. Computed on the test dataset and averaged over ten training-test splits. In both use cases, for high values of the true response (partition 10), all the methods tend to undercover. However, QR+ and CQR+ adapt the length of the prediction intervals to achieve slightly better conditional coverages. For low values of the true response (partition 1) in *protein* use case, only CV+ and J+aB manage to reach the expected coverage.

a linear regression model that predicts conditional quantiles (*linear*), implemented in *scikit-learn* [18]. The optimization is based on the pinball loss with L1 regularization. For $\hat{\mu}$, we set the quantile parameter to 0.5.

The hyper-parameters of the regression model $\hat{\mu}$ are optimized through 10-fold cross-validation with 20 iterations of a randomized search. In order to limit the computational burden, we use the same hyper-parameter values for the conditional quantile regressors $\hat{q}_{\alpha/2}$ and $\hat{q}_{1-\alpha/2}$. For a fair comparison, we follow a similar process for $\hat{r}_{\alpha/2}$ and $\hat{r}_{1-\alpha/2}$: (i) The hyper-parameters of a regression model fitted on the residual dataset are optimized through 10-fold cross-validation with 20 iterations of a randomized search; (ii) We use this same hyper-parameter configuration for the conditional quantile regressors $\hat{r}_{\alpha/2}$ and $\hat{r}_{1-\alpha/2}$. Lastly, CV+ and J+aB methods are implemented with the *mapie* package [15].

Evaluation To compare the different methods on the test data, we compute the mean *interval width* of the prediction intervals. We also evaluate the *marginal coverage* which is defined as the proportion of response values that lie within the prediction intervals.

Baselines We compare our methods to four baselines:

- *Conditional quantile regression (QR)* [11]: As the conditional quantiles estimated by quantile regressions are sometimes not well-calibrated [21], the quantile hyper-parameter is tuned using cross-validation with 10 folds in order to optimize the coverage.
- *Conformalized quantile regression (CQR)* [21]: For a fair comparison with CQR+, we implement a 10-fold cross-validation version of CQR. Therefore, the conformity scores are computed through cross-validation.
- *CV+ for K-fold cross-validation (CV+)* [2]: We use 10 folds for this method as well.
- *Jackknife+-after-bootstrap method (J+aB)* [10]: In this method, $B = 20$ training datasets are created by bootstrapping.

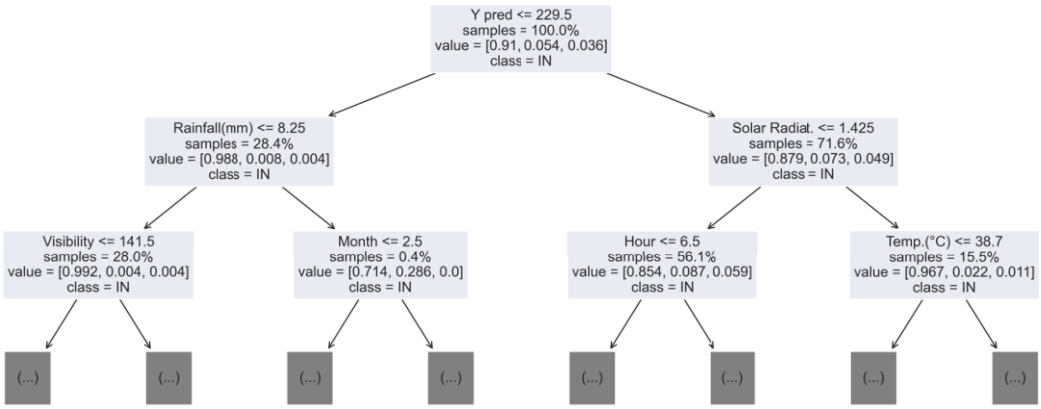

Figure 3. Undercoverage Tree Analysis (*bike* with *lgbm*) fitted on the test dataset and based on CQR+ output. The tolerance parameter δ equals 0.1%. The tree structure exhibits the regions (leaf nodes) with the proportion of samples, the proportion of labels, and the related constraints. The second leaf node from the left produces prediction intervals which significantly undercover: [0.714, 0.286, 0.0]: 71.4% is the share of samples where the prediction interval includes the true response ("IN" is the majority class) and the sum of the shares for the last two labels (0.286 and 0.) represent the miscoverage rate. In particular, 28.6% is the share of samples where the prediction intervals overestimate the true response.

Our methods We use the QR+ and CQR+ CV versions described in Algorithms 1 and 2, respectively. We set $K = 10$ folds. Similarly to QR, the quantile hyper-parameter for QR+ is tuned using cross-validation with 10 folds in order to optimize the coverage.

3.2 Results

Marginal coverage and interval width The results in Table 1 show that all methods empirically achieve roughly 90% coverage.

The results in Table 2 demonstrate that CQR+ produces the shortest intervals on average and thus outperforms CQR. Similarly, QR+ is very close to CQR+ and outperforms QR. Correcting the prediction with the predicted conditional quantiles of the residual variable seems to decrease the prediction uncertainty.

Conditional coverage and interval width The bar charts from Figure 2 evidence that QR+ and CQR+ produce adaptive prediction intervals based on a quantile-based partition of the true response. The charts display the conditional coverage and conditional interval's width, respectively. For instance, for high values of the true response, the base regression model tends to be more uncertain. Consequently, QR+ and CQR+ generate wider prediction intervals in order to achieve slightly better conditional coverages. However, in the first partition of the true response of *protein* use case, only CV+ and J+aB manage to reach the expected coverage. That is why UTA is required in order to disclose untrustworthy regions.

Undercoverage Tree Analysis We illustrate the interest of the Undercoverage Tree Analysis applied to CQR+ (CV) with the *bike* dataset. The binary tree structure is plotted in Figure 3. The left child of a node is the one which respects the split condition. To make the tree constraints more understandable, the data has not been scaled. Using the Gini impurity criterion, we fit a decision tree classifier on the test dataset with $\hat{\mu}(X)$ (Y_{pred} in the plot) as additional feature: $\hat{t} = \mathcal{T}(\{((X_j, \hat{\mu}(X_j)), U_j) : j \in \mathcal{I}_{test}\})$. δ is set to 0.1% via the parameter for the minimum number of samples in a leaf node. The second leaf node from the left displays 28.6% miscoverage (score for label 1). When the regression model predicts an hourly number of rented bikes lower than 230 and when the rainfall reaches a certain level, the model tends to output prediction intervals which exceed the true responses, with 28.6% probability. This type of analysis can also be useful for applications in the medical or financial fields where practitioners need to know whether the prediction interval is reliable.

4 Conclusion, limitations, and future work

We presented QR+ and CQR+, two methods which output prediction intervals by leveraging the estimate of a conditional quantile function of the residual variable given a specific representation. CQR+ also exploits conformal prediction to produce even shorter intervals. We presented UTA, a tree-based representation that identifies untrustworthy regions in terms of miscoverage. We have shown the relevance of these techniques on various use cases.

Our intuition to include $\hat{\mu}(X)$ as input for the conditional quantile regression is based on the residual plot: $\hat{\mu}(X)$ may be an obvious feature to detect residual singularities (as it is to visualize bias/variance from a residual plot). Further, $\hat{\mu}(X)$, as the regressor's output, may give more insights about model uncertainty than just the raw data X. However, there is no theoretical justification to demonstrate the value of $\hat{\mu}(X)$ as input for the conditional quantile regression. With the undercoverage tree analysis of QR+ or CQR+, we could select samples from leaves with high undercoverage and compare the performance with the other baselines. UTA could also be employed as a predictive tool in order to estimate, for each new input, whether the prediction interval contains the true response or not. When the true response becomes available, we could then estimate the accuracy of UTA. Further, we could experiment with additional algorithms such as random forests or neural networks (with pinball loss to estimate conditional quantiles). Additional expected coverage values could be tested as well. Lastly, we could study the efficacy of our methods when distribution shifts occur and in unstable settings. In particular, we could implement the version of CQR+ presented at the end of Subsection 2.4.

References

[1] Francis J Anscombe, 'Graphs in statistical analysis', *The american statistician*, **27**(1), 17–21, (1973).

[2] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani, 'Predictive inference with the jackknife+', *The Annals of Statistics*, **49**(1), 486–507, (2021).

[3] Thomas Bonnier and Benjamin Bosch, 'Engineering uncertainty representations to monitor distribution shifts', in *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, (2022).

[4] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[5] Dean De Cock, 'Ames, iowa: Alternative to the boston housing data as an end of semester regression project', *Journal of Statistics Education*, **19**(3), (2011).

[6] Dean De Cock. House prices - advanced regression techniques. https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data, 2011. Accessed: 2022-12-01.

[7] Lorrie Faith Cranor, 'A framework for reasoning about the human in the loop', in *Usability, Psychology, and Security, UPSEC'08, San Francisco, CA, USA, April 14, 2008, Proceedings*. USENIX Association, (2008).

[8] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani, 'The limits of distribution-free conditional predictive inference', *Information and Inference: A Journal of the IMA*, **10**(2), 455–482, (2021).

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, 'Lightgbm: A highly efficient gradient boosting decision tree', in *Advances in Neural Information Processing Systems*, eds., I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, volume 30. Curran Associates, Inc., (2017).

[10] Byol Kim, Chen Xu, and Rina Barber, 'Predictive inference is free with the jackknife+-after-bootstrap', in *Advances in Neural Information Processing Systems*, eds., H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, volume 33, pp. 4138–4149. Curran Associates, Inc., (2020).

[11] Roger Koenker and Gilbert Bassett Jr, 'Regression quantiles', *Econometrica: journal of the Econometric Society*, 33–50, (1978).

[12] Ron Kohavi, 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pp. 1137–1145, (1995).

[13] 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).

[14] Jing Lei, James Robins, and Larry Wasserman, 'Distribution-free prediction sets', *Journal of the American Statistical Association*, **108**(501), 278–287, (2013).

[15] Mapie. Mapie package. https://github.com/scikit-learn-contrib/MAPIE. Accessed: 2022-12-01.

[16] Harris Papadopoulos, 'Inductive conformal prediction: Theory and application to neural networks', in *Tools in artificial intelligence*, Citeseer, (2008).

[17] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alexander Gammerman, 'Inductive confidence machines for regression', in *Machine Learning: ECML 2002, 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002, Proceedings*, eds., Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, volume 2430 of *Lecture Notes in Computer Science*, pp. 345–356. Springer, (2002).

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., 'Scikit-learn: Machine learning in python', *the Journal of machine Learning research*, **12**, 2825–2830, (2011).

[19] Francesco Ranzato and Marco Zanella, 'Abstract interpretation of decision tree ensemble classifiers', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5478–5486, (2020).

[20] Michael Redmond and Alok Baveja, 'A data-driven software tool for enabling cooperative information sharing among police departments', *European Journal of Operational Research*, **141**(3), 660–678, (2002).

[21] Yaniv Romano, Evan Patterson, and Emmanuel Candes, 'Conformalized quantile regression', in *Advances in Neural Information Processing Systems*, eds., H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, volume 32. Curran Associates, Inc., (2019).

[22] Kamaljot Singh, Ranjeet Kaur Sandhu, and Dinesh Kumar, 'Comment volume prediction using neural networks and decision trees', in *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015)*, (2015).

[23] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz, 'Understanding failures of deep networks via robust feature extraction', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12853–12862, (2021).

[24] Ingo Steinwart and Andreas Christmann, 'Estimating conditional quantiles with the help of the pinball loss', *Bernoulli*, **17**(1), 211–225, (2011).

[25] UCI. Concrete compressive strength data set. https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength, 2007. Accessed: 2022-12-01.

[26] UCI. Communities and crime data set. https://archive.ics.uci.edu/ml/datasets/communities+and+crime, 2009. Accessed: 2022-12-01.

[27] UCI. Physicochemical properties of protein tertiary structure data set. https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure, 2013. Accessed: 2022-12-01.

[28] UCI. Facebook comment volume dataset data set. https://archive.ics.uci.edu/ml/datasets/Facebook+Comment+Volume+Dataset, 2016. Accessed: 2022-12-01.

[29] UCI. Seoul bike sharing demand data set. https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand, 2020. Accessed: 2022-12-01.

[30] Sathishkumar VE and Yongyun Cho, 'A rule-based model for seoul bike sharing demand prediction using weather data', *European Journal of Remote Sensing*, **53**(sup1), 166–183, (2020).

[31] Sathishkumar VE, Jangwoo Park, and Yongyun Cho, 'Using data mining techniques for bike sharing demand prediction in metropolitan city', *Computer Communications*, **153**, 353–366, (2020).

[32] Vladimir Vovk, 'Conditional validity of inductive conformal predictors', in *Asian conference on machine learning*, pp. 475–490. PMLR, (2012).

[33] Vladimir Vovk, 'Cross-conformal predictors', *Annals of Mathematics and Artificial Intelligence*, **74**(1), 9–28, (2015).

[34] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer, *Algorithmic learning in a random world*, Springer Science & Business Media, 2005.

[35] I-C Yeh, 'Modeling of strength of high-performance concrete using artificial neural networks', *Cement and Concrete research*, **28**(12), 1797–1808, (1998).