Tutorial on conformal prediction and related methods - ETICS 2024 Research School

Sébastien Da Veiga

Tutorial on conformal prediction & related methods

ETICS 2024 Research School

Sébastien DA VEIGA
ENSAI - CREST

September 2024

Abstract

These lecture notes have been prepared for the ETICS 2024 research school. They consist of two 3-hours lectures, dedicated to the following topics:

- Introduction & motivation for uncertainty quantification on machine learning predictions
- Conformal prediction theory & methodology
- Extensions & concurrent methods

All figures and examples from these notes can be reproduced with the R code available here: CP tutorial material. Accompanying notebooks in R and Python can also be found there. Any remark or suggestion concerning this tutorial can be sent to sebastien.da-veiga@ensai.fr .

I took inspiration from great lecture notes and tutorials available online, and I strongly encourage you to read them if you want to go further: see for example Angelopoulos and Bates (2021), Tibshirani (2024) and Barber (2024).

Contents

Lecture 1: Introduction to conformal prediction

Day 1

- Motivation
- Preliminaries
  - Reminder on quantiles and exhangeability
  - Elementary results on ranks and order statistics
- Conformal prediction theory
  - Split conformal prediction
  - Resampling strategies: jackknife, jackknife+, CV+
  - Full conformal prediction
- Summary and introduction to lecture 2

Chapter 1

Uncertainty quantification on machine learning predictions

1.1 Problem setting

In supervised statistical / machine learning, we are given a sample $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\dots,n}$ of size n from an unknown joint distribution $P_{\boldsymbol{X}Y}$ defined on a product space $\mathcal{X} \times \mathcal{Y}$. Here we will focus on the specific regression setting where $\boldsymbol{X} = \left(X^{(1)}, \dots, X^{(d)}\right) \in \mathcal{X} \subset \mathbb{R}^d$ is the vector of the explanatory variables or inputs or *features* and $Y \in \mathcal{Y} \subset \mathbb{R}$ is the *target* variable, or output. We make this choice since it is the most common in computer experiments, but almost all the material discussed in this course can be readily extended to other frameworks, and in particular the classification setting where $\mathcal{Y} = \{C_1, \dots, C_M\}$ with C_j, $j = 1, \dots, M$ being M distinct categories.

In this setting, the traditional goal is to build a map $\hat{\mu}_{\mathcal{D}_n} : \mathbb{R}^d \mapsto \mathbb{R}$, called a *predictor* or *prediction function* or *prediction rule*, from the observations in the sample $\mathcal{D}_n$ and which predicts the value of the target variable $\hat{y} = \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x})$ of any other individual with features $\boldsymbol{x}$. Note that we explicitly write the dependence between the predictor $\hat{\mu}_{\mathcal{D}_n}$ and the sample $\mathcal{D}_n$ used to build it, since this notation will be useful later on. You certainly know a substantial number of different methods to build such a predictor, among which

- linear or polynomial regression
- nearest-neighbor and kernel smoothing methods, or more generally local-averaging techniques
- smoothing splines and RKHS techniques, or Bayesian variants such as Gaussian process regression
- ensemble methods such as random forests or boosting
- (deep) neural networks

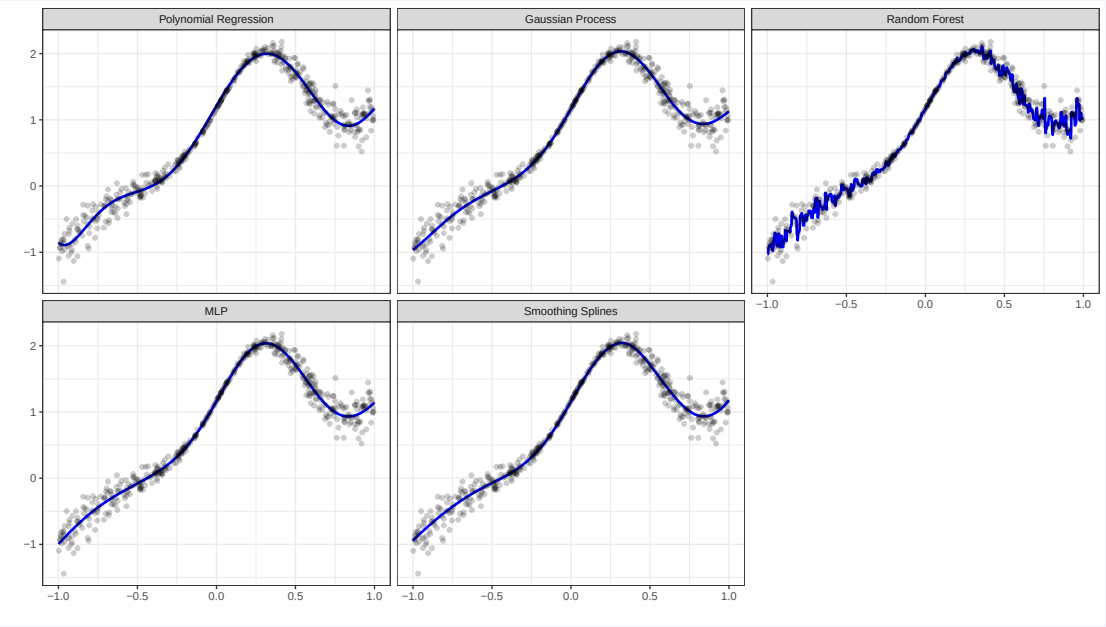
- {insert your favorite method}

Let us first display some of them on a simple one-dimensional regression example, that will serve as an illustration all along this course.

Example. We consider the analytical one-dimensional test case defined as

$$Y = X^3 + 2\exp(-6(X - 0.3)^2) + \varepsilon$$

where $X \sim \mathcal{U}[-1, 1]$ and $\varepsilon \sim \mathcal{N}(0, 0.2|X|)$ taken from `https://www.tidymodels.org/learn/models/conformal-regression/`. Note that due to the definition of ε, this is an heteroskedastic model. We generate a training dataset $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\ldots,n}$ of size $n = 500$, and train five regression models: a polynomial regression, a Gaussian process, a multi-layer perceptron, a random forest and a smoothing spline. Predictions for each of these models on a grid of 1000 equally spaced test points are displayed below (training data as black points and predictions in blue).



Except for the random forest model, observe that all models provide very similar predictions on the test set (keep this in mind for what follows!).

Now, in practice, **just providing model predictions is no longer sufficient** in this era where machine learning is extensively used in industrial and high-stake fields: we also need to build **prediction intervals** around the predictions, which should contain the true but unobserved value of the target with a certain level of confidence. Without being exhaustive, here are a few benefits we can expect by producing prediction intervals:

- building *trust* with the end-users of the machine learning model, since they will be able to see how confident it is when making a prediction
- opening the path for *sequential or adaptive design of experiments*, by requesting new labeled data in the feature regions where the model is not confident (a.k.a. *active learning*)
- detecting potential out-of-distribution data, if the intervals are unexpectedly wide

The strong interest in providing such prediction intervals is not at all new, and it is actually quite easy to generate some kind of intervals for some models, with different underlying ideas:

- explicit central limit theorems that are available for some models (polynomial regression, local-averaging methods, ...)
- quantile regression, where the model is trained to specifically learn quantiles of the target conditional distribution instead of the mean
- the Bayesian paradigm (Gaussian processes being a major representer of this class of methods in the computer experiment community, but we can also cite Bayesian neural networks more recently)
- resampling methods (bootstrap, cross-validation, leave-one-out, jackknife, ...)
- heuristic approaches, with a lot of popularity in the neural network community (multi-start optimization of the model loss function - e.g. deep ensembles, randomization inside the model - e.g. drop-out, ...)

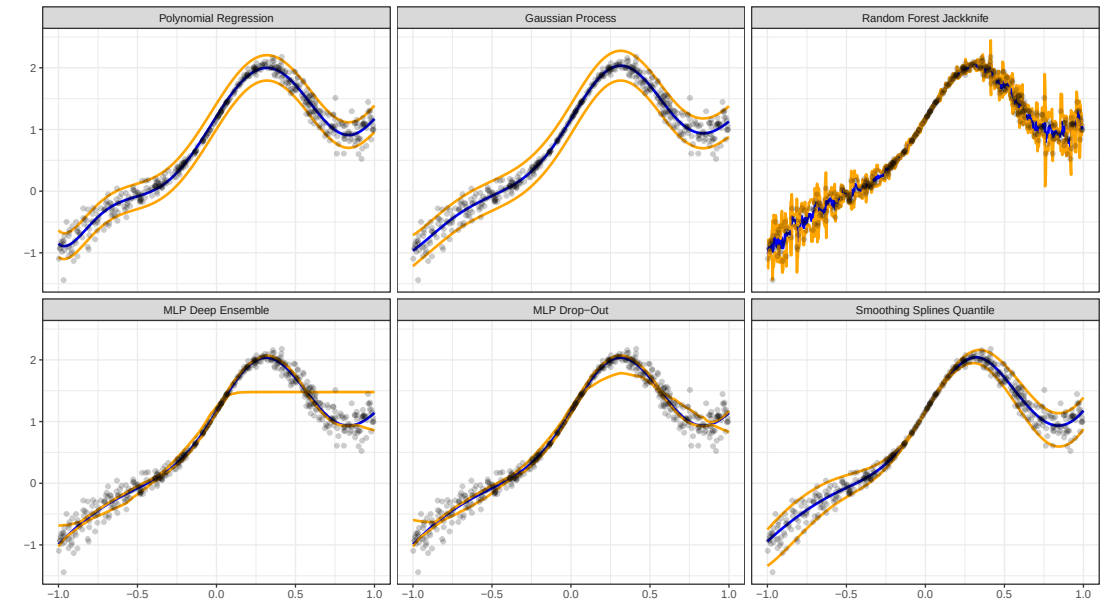
We illustrate some of them on the previous example.

Example. We consider the same analytical one-dimensional test case from before, and in addition to the model predictions, we also build prediction intervals with some of the ideas listed above:

- for the polynomial regression model, we use the textbook central limit theorem for predictions
- for Gaussian processes, we use the posterior distribution
- for random forests, we use the jackknife + out-of-bag resampling method
- for the multi-layer perceptron, we use deep ensembles and drop-out

- for smoothing splines, they are now built to estimate quantiles

For each method we compute 90% intervals around the predictions, they are represented with orange lines below (the predictions are still in blue).



Contrary to the previous prediction-only case, now the intervals produced by each method largely differ, except for the first two ones.

From this illustration, a few facts are particularly clear. First, even if model predictions can be very close, the intervals may heavily vary depending on the underlying assumptions used to build them. Second, even for a fixed model (e.g. multi-layer perceptron in the example), depending on the uncertainty quantification methodology the intervals can also change. This is highly problematic for our quest for trustworthiness mentioned before. The main question actually is: what went wrong in what we did? A partial answer lies in the fact that:

- for parametric approaches such as polynomial regression, our model may be wrong (true relationship being polynomial and homoskedastic Gaussian noise), or guarantees on the validity of intervals are only asymptotic (central limit theorem)
- for nonparametric approaches like splines, smoothness assumptions may be violated
- for Bayesian approaches, the influence of the prior is not negligible (e.g. a stationary kernel in Gaussian processes)
- for resampling approaches, we may lack theoretical guarantees that they provide valid intervals
- for heuristic approaches, the theoretical guarantees may be even more lacking

Mathematically, we actually aim at building a prediction interval defined as follows.

Definition 1 (Prediction interval/band). A **prediction interval** $\hat{C}_{\mathcal{D}_n}$ with error level $\alpha \in (0,1)$ is a function

$$\hat{C}_{\mathcal{D}_n} : \mathcal{X} \to \{\text{subsets of } \mathcal{Y}\}$$

built from an i.i.d. sample $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\dots,n}$ from $P_{\boldsymbol{X}Y}$ such that, for a new i.i.d. pair $(\boldsymbol{X}_{n+1}, Y_{n+1}) \sim P_{\boldsymbol{X}Y}$, we have

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha, \tag{1.1}$$

where **the probability is over all data** $(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$.

From this definition, notice that we target two specific properties for a prediction interval:

1. It must be **distribution-free**, i.e. the coverage guarantee (1.1) must hold without assumptions on the data generating process $P_{\boldsymbol{X}Y}$
2. It must be valid in a **non-asymptotic** framework, i.e. for any number n of samples used to build it

This is exactly what conformal prediction does, as we will see in what follows. However, without spoiling too much, we will also discuss variants that may have more practical interest for us (from a theoretical and computational viewpoint), but will come at the cost of (slightly) relaxing the above properties.

Remark. Observe that in the definition above:

- since the probability is over all data, this means that the coverage is guaranteed in average over all random draws of training data (used to build the prediction interval) and random draws of testing data (where we predict)
- the testing data is supposed here to follow the same distribution as the training data, meaning that we will have to make adjustments to handle important practical situations (e.g. timeseries, active learning, ...)

We will comment on these facts later in this course.

But before diving into the details of conformal prediction, we will first need some notations and reminder of results on quantiles and exchangeability that will be useful later on.

1.2 Preliminaries

We start with the definition of quantile which we will use hereafter.

Definition 2 (Sample quantile). For a quantile level $\tau \in [0,1]$ and a list of samples $z_1, \dots, z_n \in \mathbb{R}$ of size n, the τ-quantile is

$$\text{Quantile}_\tau\left(z_1, \dots, z_n\right) = \text{Quantile}_\tau\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{z_i}\right) := z_{(\lceil n\tau \rceil)}$$

where $z_{(i)}$ is the ith order statistic of $z_1, \dots, z_n$. In words, the τ-quantile is the $\lceil n\tau \rceil$th smallest value of $z_1, \dots, z_n$. This is also equivalent to

$$\text{Quantile}_\tau\left(z_1, \dots, z_n\right) = \inf\left\{t \in \mathbb{R} : \ \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{z_i \leq t} \geq \tau\right\}$$

for $\tau \geq 1/n$, the empirical counterpart of the population quantile

$$\text{Quantile}_\tau\left(Z\right) = \inf\left\{t \in \mathbb{R} : \ \mathbb{P}(Z \leq t) \geq \tau\right\}$$

where $\mathbb{P}(Z \leq t) = F_Z(t)$ is the cumulative distribution function of Z.

Remark (Convention). Observe that we use here the convention for quantiles corresponding to the inverse of the empirical distribution function. See `https://en.wikipedia.org/wiki/Quantile#Estimating_quantiles_from_a_sample` for a quite exhaustive list of other possible choices.

Remark (Notation). We introduced above two alternate notations, and in particular the second one

$$\text{Quantile}_\tau\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{z_i}\right)$$

where the empirical distribution function appears explicitly. It will be useful later when considering *weighted* samples, in which case we will write

$$\text{Quantile}_\tau\left(\sum_{i=1}^{n} w_i\delta_{z_i}\right)$$

for positive weights $w_1, \dots, w_n$ summing to 1, the τ-quantile still being defined as the inverse of the (weighted) empirical distribution function.

Another central ingredient for conformal prediction is the concept of exchangeable random variables.

Definition 3 (Exchangeability). Random variables $Z_1, \ldots, Z_n$ are **exchangeable** if

$$(Z_1, \ldots, Z_n) \overset{\mathrm{d}}{=} \left(Z_{\sigma(1)}, \ldots, Z_{\sigma(n)}\right)$$

for every finite permutation σ of the indices $1, \ldots, n$.

Four important practical cases for us lead to exchangeable sequences.

Proposition 1 (Particular cases of exchangeable sequence). The random variables $Z_1, \ldots, Z_n$ are exchangeable in the following cases:

(i) $Z_1, \ldots, Z_n$ are i.i.d. from some distribution P

(ii) $Z_1, \ldots, Z_n | \theta$ are i.i.d. from some distribution $P(\cdot|\theta)$ indexed by some random vector $\theta \sim P_\theta$ (*conditional i.i.d*)

(iii) $Z_1, \ldots, Z_n$ are sampled uniformly without replacement from a finite set

(iv) $Z_1 = f(W_1), \ldots, Z_n = f(W_n)$ with $W_1, \ldots, W_n$ exchangeable and f any function

Proposition 2 (A simple but useful result on quantiles). For any $z_1, \ldots, z_n, t \in \mathbb{R}$ and quantile level $\tau \in [0, 1]$, we have

$$t \leq \text{Quantile}_\tau\left(z_1, \ldots, z_n, t\right) \Leftrightarrow t \leq \text{Quantile}_{\tau\frac{n+1}{n}}\left(z_1, \ldots, z_n\right)$$

Proof. We will prove the equivalence between the complementary of these events. $t > \text{Quantile}_\tau\left(z_1, \ldots, z_n, t\right)$ is equivalent to $t >$ $\lceil (n+1)\tau \rceil$th smallest value of $z_1, \ldots, z_n, t$ by definition of the τ-quantile of the samples $z_1, \ldots, z_n, t$. Since t cannot been larger than itself, this is also equivalent to $t > \lceil (n+1)\tau \rceil$th smallest value of $z_1, \ldots, z_n$, that is $t > \lceil n\tau\frac{n+1}{n} \rceil$th smallest value of $z_1, \ldots, z_n$ and the latter is by definition the $\tau\frac{n+1}{n}$-quantile of the samples $z_1, \ldots, z_n$. □

Proposition 3 (A central result for exchangeable sequences). If $Z_1, \ldots, Z_n$ are exchangeable, then $\forall i = 1, \ldots, n$

$$\mathbb{P}\left(Z_i \leq \text{Quantile}_\tau\left(Z_1, \ldots, Z_n\right)\right) \geq \tau$$

for any quantile level $\tau \in [0, 1]$.

Proof. By exchangeability, the rank of Z_i is uniformly distributed over $1, \ldots, n$, and the result follows. □

1.3 A simple but illustrative example

Let us start with an extremely simplified situation, where we do not have features at all and only observe an i.i.d. sample $\mathcal{D}_n = (Y_1, \dots, Y_n) \sim P$, from which we want to build a prediction interval for a new sample $Y_{n+1} \sim P$. Focusing on a one-sided interval $\hat{C}_{\mathcal{D}_n} = (-\infty, \hat{q}_n)$, we thus seek $\hat{q}_n$ such that

$$\mathbb{P}\left(Y_{n+1} \leq \hat{q}_n\right) \geq 1 - \alpha.$$

Remembering Definition 2 gives a strong hint that $\hat{q}_n$ should be chosen as a $(1-\alpha)$-quantile of some sort, but which one? There are two naive ways to tackle this problem:

1. We could make a distributional assumption on P, for example that it is a Gaussian with unknown mean and variance. As an illustration, denoting $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n Y_i$ the empirical mean and $s_n^2 = \frac{1}{n-1}\sum_{i=1}^n \left(Y_i - \bar{Y}_n\right)^2$ the empirical variance, in this case we know that

$$\frac{Y_{n+1} - \bar{Y}_n}{s_n\sqrt{1+1/n}} \sim T^{n-1},$$

   a Student's t-distribution with $n-1$ degrees of freedom, meaning that the choice

$$\hat{q}_n = \bar{Y}_n + s_n\sqrt{1+1/n}\,\text{Quantile}_{1-\alpha}\left(T^{n-1}\right)$$

   satisfies the targeted coverage (with an equality) for any finite n. Unfortunately this does not lead to a distribution-free interval, since it is only valid under the Gaussian assumption we did.

2. On the opposite, we could adopt a non-parametric approach by simply using

$$\hat{q}_n = \text{Quantile}_{1-\alpha}\left(Y_1, \dots, Y_n\right)$$

   the empirical quantile of the observations. Unfortunately this would only give an approximate coverage

$$\mathbb{P}\left(Y_{n+1} \leq \hat{q}_n\right) \approx 1 - \alpha,$$

   since the exact one would be obtained with the population quantile of P, and the empirical quantile above only converges towards this grail when $n \to \infty$ under classical assumptions. In other words, this time we have a distribution-free interval, but which is only valid when $n \to \infty$.

None of these approaches thus solve our problem, but now come into play the results on quantiles and exchangeable variables recapped before! Indeed, Proposition 3 almost corresponds to what we want: $Y_1, \dots, Y_n, Y_{n+1}$ are exchangeable since they are i.i.d. (Proposition 1 (i)), so we have

$$\mathbb{P}\left(Y_{n+1} \leq \text{Quantile}_{1-\alpha}\left(Y_1, \dots, Y_n, Y_{n+1}\right)\right) \geq 1 - \alpha \tag{1.2}$$

by applying the proposition for $i = n+1$ and $\tau = 1-\alpha$. We cannot conclude yet because taking

$$\hat{q}_n = \text{Quantile}_{1-\alpha}\left(Y_1, \dots, Y_n, Y_{n+1}\right)$$

is not possible since it depends on the unknown value Y_{n+1}. Fortunately, a miracle appears thanks to Proposition 2, which states that the following events are equivalent:

$$Y_{n+1} \leq \text{Quantile}_{\tau}\left(Y_1, \ldots, Y_n, Y_{n+1}\right) \Leftrightarrow Y_{n+1} \leq \text{Quantile}_{\tau\frac{n+1}{n}}\left(Y_1, \ldots, Y_n\right).$$

This means that we can rewrite Equation (1.2) as

$$\mathbb{P}\left(Y_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(Y_1, \ldots, Y_n\right)\right) \geq 1-\alpha,$$

and we can finally choose

$$\hat{q}_n = \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(Y_1, \ldots, Y_n\right)$$

which now only depends on the observed samples $Y_1, \ldots, Y_n$ and directly provides the coverage guarantee for any n and with no distribution assumption on P. This can be thought of as a finite-sample correction of the empirical quantile, where the level is adjusted so that the coverage guarantee holds for any n.

Remark. If we assume that there is almost surely no ties between $Y_1, \ldots, Y_n$ (e.g. if we assume that P is absolutely continuous), we actually have a stronger statement with an upper bound:

$$1-\alpha+\frac{1}{n+1} \geq \mathbb{P}\left(Y_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(Y_1, \ldots, Y_n\right)\right) \geq 1-\alpha,$$

see Tibshirani (2024) for example.

1.4 Conformal prediction for supervised learning

After this simplified example, let us now see if we can directly apply the same ideas in the supervised setting, where we observe an i.i.d. sample $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\ldots,n}$ from $P_{\boldsymbol{X}Y}$, and want to build a prediction interval for Y_{n+1} as a function of $\boldsymbol{X}_{n+1}$, for an i.i.d. pair $(\boldsymbol{X}_{n+1}, Y_{n+1}) \sim P_{\boldsymbol{X}Y}$.

As mentioned at the beginning, we typically want a prediction interval centered around a point prediction given by $\hat{\mu}_{\mathcal{D}_n}$, a predictor which has been trained on $\mathcal{D}_n$, which means that we seek a prediction interval of the form $\hat{\mu}_{\mathcal{D}_n}(X_{n+1}) \pm \ldots$, that is for example $\hat{C}_{\mathcal{D}_n}(\boldsymbol{x}) = [\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) - \hat{q}_n, \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + \hat{q}_n]$ or $\hat{C}_{\mathcal{D}_n}(\boldsymbol{x}) = \{y \in \mathbb{R} : \ |y - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x})| \leq \hat{q}_n\}$. Using the last equality, the targeted coverage is thus

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) = \mathbb{P}\left(|Y_{n+1} - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})| \leq \hat{q}_n\right) = \mathbb{P}\left(R_{n+1} \leq \hat{q}_n\right) \geq 1-\alpha \qquad (1.3)$$

where we denote $R_i = |Y_i - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_i)|$ the absolute residuals for $i = 1, \ldots, n+1$. Introducing here the residuals is deliberate, because the last part in Equation (1.3) is identical to the previous

simple problem, with $Y_1, \ldots, Y_n, Y_{n+1}$ replaced by the residuals $R_1, \ldots, R_n, R_{n+1}$. As before, we can then take

$$\hat{q}_n = \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(R_1, \ldots, R_n\right)$$

and the course is finished.

Obviously it is not, because using this quantile in the prediction interval will not lead to the expected coverage! It would be valid if $R_1, \ldots, R_n, R_{n+1}$ were exchangeable, but unfortunately they are not. Indeed, $R_1, \ldots, R_n$ are the residuals of the prediction model on the training set ($\hat{\mu}_{\mathcal{D}_n}$ was trained on $\mathcal{D}_n$ by definition), whereas R_{n+1} corresponds to the error of the prediction model on an unseen testing point: the latter will thus be generally much larger than the former. This is similar to the classical phenomenon in supervised learning where the empirical risk computed on the training set is not representative of the prediction error computed on a test set.

We will see in this section three different ways to address this.

1.4.1 Split conformal prediction

Keeping in mind the similarity with the empirical risk in supervised learning just mentioned, you may easily think about a trick to solve the exchangeability problem in Equation (1.3). Indeed the problem comes from the fact that the quantile is computed on the training set $\mathcal{D}_n$ while R_{n+1} is a residual on the test set, resulting on residuals that do not compare with each other. A straightforward fix thus consists in computing the quantile on data that were not used for training the predictor. To achieve this, as is done in supervised learning, we can split the training set $\mathcal{D}_n = \mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{cal}}$ in two disjoint sets:

- $\mathcal{D}_n^{\text{ptrain}}$ is the *proper training* set used to build the predictor $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$ only (in conformal prediction it may sometimes be referred to as the *pretraining* set)
- $\mathcal{D}_n^{\text{cal}}$ is the hold-out *calibration set* on which the residuals are computed, which has the same role as the validation set in supervised learning

This time, by computing the quantile of the residuals on $\mathcal{D}_n^{\text{cal}}$, we can now achieve the required coverage. To see this, let us consider the absolute residuals $R_i = |Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)|$ for $i \in \mathcal{D}_n^{\text{cal}}$ on the calibration set and R_{n+1} the test residual. Of course these residuals are not independent since they all depend on $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$ and thus on $\mathcal{D}_n^{\text{ptrain}}$. However, when we condition on $\mathcal{D}_n^{\text{ptrain}}$, they obviously become i.i.d., meaning that **they are exchangeable** by Proposition 1 (ii). This leads directly to the following result:

$$\mathbb{P}\left(R_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n_{\text{cal}}+1}{n_{\text{cal}}}}\left(R_i,\ i \in \mathcal{D}_n^{\text{cal}}\right) | \mathcal{D}_n^{\text{ptrain}}\right) \geq 1 - \alpha \tag{1.4}$$

or

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{ptrain}}\right) \geq 1 - \alpha$$

for a prediction interval of the form

$$\hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \hat{q}_{\mathcal{D}_n^{\text{cal}}}$$

where $\hat{q}_{\mathcal{D}_n^{\text{cal}}} = \text{Quantile}_{(1-\alpha)\frac{n_{\text{cal}}+1}{n_{\text{cal}}}}\left(R_i,\ i \in \mathcal{D}_n^{\text{cal}}\right)$ and n_{cal} is the size of the calibration set. Such a procedure is called **split conformal prediction**, *split* obviously referring to the fact that we separated the training set in a proper training set and a calibration set. For later reference, we formalize this result in a theorem.

Theorem 1 (Coverage for split conformal - Vovk et al. (2005)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$, $(\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable, the split conformal interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{ptrain}}\right) \geq 1 - \alpha.$$

Observe finally that here the coverage guarantee is conditional on the proper training set $\mathcal{D}_n^{\text{ptrain}}$, meaning that the probability is over data $\{(\boldsymbol{X}_i, Y_i)\}_{i \in \mathcal{D}_n^{\text{cal}}}, (\boldsymbol{X}_{n+1}, Y_{n+1})$, which differs from the coverage from Equation (1.1). We discuss this point in the following remark.

Remark (Marginal vs training conditional coverage). It is straightforward to see that, by marginalizing over the proper training set, Equation (1.4) becomes

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) = \mathbb{E}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{ptrain}}\right)\right\} \geq 1 - \alpha$$

where now the probability is over all data $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ as in Equation (1.1). This type of guarantee is called **marginal coverage**, in the sense that the probability has been marginalized over all the randomness. Split conformal prediction thus satisfies also a marginal coverage guarantee.

At the opposite, we may look at the coverage where we only marginalize over the test set, i.e.

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n\right),$$

which is called the **training conditional** coverage and split conformal prediction also comes with this kind of guarantee! Indeed, we can show that for split conformal

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n\right) \sim \text{Beta}(l, n_{\text{cal}} + 1 - l)$$

where $l = \lceil (1 + n_{\text{cal}})(1 - \alpha) \rceil$ by classical results on the distribution of order statistics (under the assumption that there are no ties in the residuals), see Angelopoulos and Bates (2021) or Tibshirani (2024) for illustrations.

We thus deduce that the moments of the training conditional coverage satisfy

$$\begin{aligned}
\mathbb{E}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})|\mathcal{D}_n\right)\right\} &= \tfrac{\lceil (1+n_{\text{cal}})(1-\alpha)\rceil}{1+n_{\text{cal}})} \in \left[1-\alpha, 1-\alpha+\frac{1}{n_{\text{cal}}+1}\right] \\
\text{Var}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})|\mathcal{D}_n\right)\right\} &= \tfrac{l(n_{\text{cal}}+1-l)}{(n_{\text{cal}}+1)^2(n_{\text{cal}}+2)} \approx \frac{\alpha(1-\alpha)}{n_{\text{cal}}+2}
\end{aligned}$$

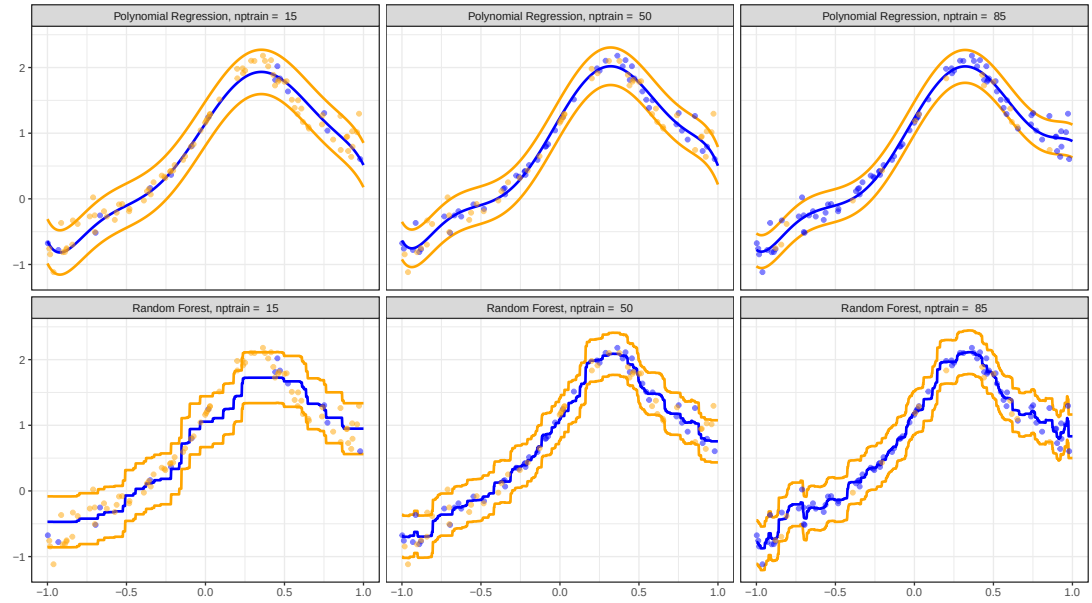
where the last approximation is valid when n_{cal} is sufficiently large. This means that n_{cal}, the number of calibration samples, plays a major role on how close the training conditional coverage is to its mean, which itself is close to $1-\alpha$. More precisely, a more quantitative result states that (Vovk, 2012)

$$\mathbb{P}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})|\mathcal{D}_n\right) \geq 1-\alpha-\sqrt{\frac{\log(1/\delta)}{2n_{\text{cal}}}}\right\} \geq 1-\delta.$$

All in one, n_{cal} should be chosen as large as possible in order to have a training conditional coverage close to $1-\alpha$ (but remember that marginal coverage will hold for any value of n_{cal}). However, this means that n_{ptrain} the size of the proper training set will then be smaller, thus leading to a degraded prediction function $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$, and ultimately to larger and thus less informative prediction intervals in general. Split conformal prediction fundamentally comes with such a trade-off, and we will see later other variants which overcomes this limitation (at the cost of additional assumptions, or other types of limitations).

We can finally show split conformal prediction in action on our running example.

Example. For our previous one-dimension test case, we take a small training dataset $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\ldots,n}$ of size $n = 100$ for better illustration. To perform split conformal prediction, we consider three scenarios, where we take successively $n_{\text{ptrain}} = 15$, 50, 85 corresponding to $n_{\text{cal}} = 85$, 50, 15. Since almost all models provided similar predictions before, we focus on just two of them, polynomial regression and random forests. For each scenario and each model we give below the predictions in blue and split conformal intervals for $\alpha = 0.1$ in orange. We also represent proper training samples from $\mathcal{D}_n^{\text{ptrain}}$ in blue, and calibration samples from $\mathcal{D}_n^{\text{cal}}$ in orange.



As expected, we can observe that the width of the intervals tend to shrink when n_{ptrain} increases.

We can also check on an independent test set if the target coverage is attained (we are actually computing the training conditional coverage, since here the training data is fixed and we only average over test data):

Poly. Regr. $n_{\text{ptrain}} = 15$	Poly. Regr. $n_{\text{ptrain}} = 50$	Poly. Regr. $n_{\text{ptrain}} = 85$	RF $n_{\text{ptrain}} = 15$	RF $n_{\text{ptrain}} = 50$	RF $n_{\text{ptrain}} = 85$
0.933	0.918	0.931	0.869	0.943	0.955

which is in line with the theoretical coverage up to statistical fluctuations (from the size of the test set and the calibration set, if you remember the discussion above).

The trade-off implied by splitting the training set $\mathcal{D}_n$ that we discussed previously is not at all specific to split conformal prediction, since it is central in supervised learning. But you know that there are prominent workarounds to retaining an hold-out set, based on resampling schemes such as the jackknife and cross-validation. Similarly, conformal prediction can also use such ideas: this is what we discuss in the next section.

1.4.2 Cross-validation+ and jackknife+ conformal prediction

Foreword We use here the term jackknife, but if you have never heard it, you may be more familiar with the terminology *leave-one-out* cross-validation, both mean the same.

We start by introducing notations following Barber et al. (2021*b*):

Definition 4 (Shortcut for quantiles). For a quantile level $\alpha \in [0,1]$ and a list of samples $z_1, \dots, z_n \in \mathbb{R}$ of size n we denote

- $\hat{q}^+_{n,\alpha}\{z_i\} = z_{\lceil (1-\alpha)(n+1) \rceil}$ the sample $(1-\alpha)\frac{n+1}{n}$-quantile of $z_1, \dots, z_n$
- $\hat{q}^-_{n,\alpha}\{z_i\} = z_{\lfloor \alpha(n+1) \rfloor}$, the sample $\alpha\frac{n+1}{n}$-quantile of $z_1, \dots, z_n$ up to a different rounding

with sample quantiles defined in Definition 2.

We will also need two trivial properties:

Proposition 4. With the notations above, we have

(i) $\hat{q}^-_{n,\alpha}\{z_i\} = -\hat{q}^+_{n,\alpha}\{-z_i\}$

(ii) $\forall a \in \mathbb{R}$, $\hat{q}^+_{n,\alpha}\{a + z_i\} = a + \hat{q}^+_{n,\alpha}\{z_i\}$ and $\hat{q}^-_{n,\alpha}\{a + z_i\} = a + \hat{q}^-_{n,\alpha}\{z_i\}$

We discussed before that the naive prediction interval

$$\hat{C}^{\text{naive}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) \pm \hat{q}^+_{n,\alpha}\{|Y_i - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_i)|,\ i = 1, \dots, n\}$$

does not have coverage guarantee since the residuals on the training set involved in the quantile are not comparable to the residuals we expect on a test point. Borrowing ideas from the traditional jackknife in supervised learning, we may then think about using leave-one-out residuals in the quantile computation instead, since they are representative of the errors on a test point:

$$\hat{C}^{\text{jack}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) \pm \hat{q}^+_{n,\alpha}\{|Y_i - \hat{\mu}_{-i}(\boldsymbol{X}_i)|,\ i = 1, \dots, n\} \tag{1.5}$$

where $\hat{\mu}_{-i}$ denotes the predictor trained on the data $\mathcal{D}_n \setminus i$. Unfortunately, this straightforward jackknife interval does not have theoretical coverage guarantee without additional assumptions (and thus counterexamples exist, but see the remark below), even if in practice it appears to be the case (Barber et al., 2021*b*). However, rewriting such an interval differently gives a hint on the modifications we could make to solve this problem:

$$\begin{aligned}
\hat{C}^{\text{jack}}_{\mathcal{D}_n}(\boldsymbol{x}) &= \big[\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) - \hat{q}^+_{n,\alpha}\{|Y_i - \hat{\mu}_{-i}(\boldsymbol{X}_i)|,\ i = 1, \dots, n\}, \\
&\qquad \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + \hat{q}^+_{n,\alpha}\{|Y_i - \hat{\mu}_{-i}(\boldsymbol{X}_i)|,\ i = 1, \dots, n\}\big] \\
&= \big[\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) - \hat{q}^+_{n,\alpha}\{R^{\text{LOO}}_i\}, \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + \hat{q}^+_{n,\alpha}\{R^{\text{LOO}}_i\}\big] \\
&= \big[\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + \hat{q}^-_{n,\alpha}\{-R^{\text{LOO}}_i\}, \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + \hat{q}^+_{n,\alpha}\{R^{\text{LOO}}_i\}\big] \quad \text{Proposition 4, (i)} \\
&= \big[\hat{q}^-_{n,\alpha}\{\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) - R^{\text{LOO}}_i\}, \hat{q}^+_{n,\alpha}\{\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) + R^{\text{LOO}}_i\}\big] \quad \text{Proposition 4, (ii)}
\end{aligned}$$

where $R^{\text{LOO}}_i = |Y_i - \hat{\mu}_{-i}(\boldsymbol{X}_i)|$, $i = 1, \dots, n$ denote the leave-one-out residuals.

With this expression, it is thus natural to think about replacing the common centering $\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x})$ in the quantiles by, again, the leave-one-out predictor $\hat{\mu}_{-i}(\boldsymbol{x})$. This leads to the so-called **jackknife+ conformal interval**:

$$\hat{C}^{\text{jack+}}_{\mathcal{D}_n}(\boldsymbol{x}) = \big[\hat{q}^-_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) - R^{\text{LOO}}_i\}, \hat{q}^+_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) + R^{\text{LOO}}_i\}\big]\,,$$

with the following coverage guarantee.

Theorem 2 (Coverage for jackknife+ conformal - Barber et al. (2021*b*)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$, $(\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable and the prediction algorithm is symmetric, the jackknife+ interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack}+}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - 2\alpha.$$

Remark. We should comment on two points appearing in the previous theorem:

- This is the first time we mention the concept of a *symmetric prediction algorithm*. This simply means that it treats the training data symmetrically, i.e. the prediction function $\hat{\mu}$ it outputs is stable under any permutation of the training data
- With jackknife+, the attained coverage is $1 - 2\alpha$ instead of the target $1 - \alpha$, but it is observed in practice to be close to the target (Barber et al., 2021*b*)

Concerning the last point, with additional assumptions related to algorithmic stability (which we will discuss during the second lecture), it is possible to get jackknife+ coverage closer to $1-\alpha$. Interestingly, under such assumptions, the straightforward jackknife interval in (1.5) now also has similar coverage guarantees.

Finally, notice that in general the computational cost of leave-one-out may be quite large (except for specific prediction algorithms such as linear regression or kernel ridge regression for example), and thus a variant based on cross-validation with K folds may be preferred. This is totally doable by following the exact same principle of the jackknife+, leading to the **CV+ conformal interval**. Define K folds $\bigcup_{k=1}^{K} \mathcal{D}_k = \mathcal{D}_n$ and denote $R^{\text{CV}} = \{|Y_i - \hat{\mu}_{-\mathcal{D}_k}(\boldsymbol{X}_i)|,\ i \in \mathcal{D}_k\}_{k=1}^{K}$ and $\hat{\mu}^{\text{CV}}(\boldsymbol{x}) = \{\hat{\mu}_{-\mathcal{D}_k}(\boldsymbol{x}),\ i \in \mathcal{D}_k\}_{k=1}^{K}$ the collection of all K-fold absolute residuals and predictions, the CV+ conformal interval is given by

$$\hat{C}^{\text{CV}+}_{\mathcal{D}_n}(\boldsymbol{x}) = \left[\hat{q}^{-}_{n,\alpha}\{\hat{\mu}^{\text{CV}}_i(\boldsymbol{x}) - R^{\text{CV}}_i\}, \hat{q}^{+}_{n,\alpha}\{\hat{\mu}^{\text{CV}}_i(\boldsymbol{x}) + R^{\text{CV}}_i\}\right].$$

It satisfies the following coverage guarantee.

Theorem 3 (Coverage for CV+ conformal - Barber et al. (2021*b*)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$, $(\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable and the prediction algorithm is symmetric, the K-fold CV+ interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{CV}+}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - 2\alpha - \min\left\{\frac{2(1-1/K)}{n/K+1}, \frac{1-K/n}{K+1}\right\}.$$

Remark. When $K = n$ we recover the jackknife+ result, and get a coverage close to $1-2\alpha$

when $K \ll n$ is small. For any value of K, we actually have

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\mathrm{CV+}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - 2\alpha - \sqrt{2/n}.$$

Recently it has been shown that CV+ intervals also come with training conditional coverage which depends on the number of samples in each fold:

Theorem 4 (Training conditional coverage for CV+ - Bian and Barber (2023)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable, the K-fold CV+ interval with each fold of size m such that $n = Km$ satisfies

$$\mathbb{P}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\mathrm{CV+}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n\right) \geq 1 - 2\alpha - \sqrt{\frac{2\log(K/\delta)}{m}}\right\} \geq 1 - \delta.$$

Remark. First note that, contrary to the marginal coverage of CV+ previously, assuming that the prediction algorithm is symmetric is no longer needed.

In addition, this theorem states that in essence if the number of samples in each fold m is large, we get training conditional coverage close to $1 - 2\alpha$. m here actually plays the same role as n_{cal} in the training conditional guarantee of split conformal.

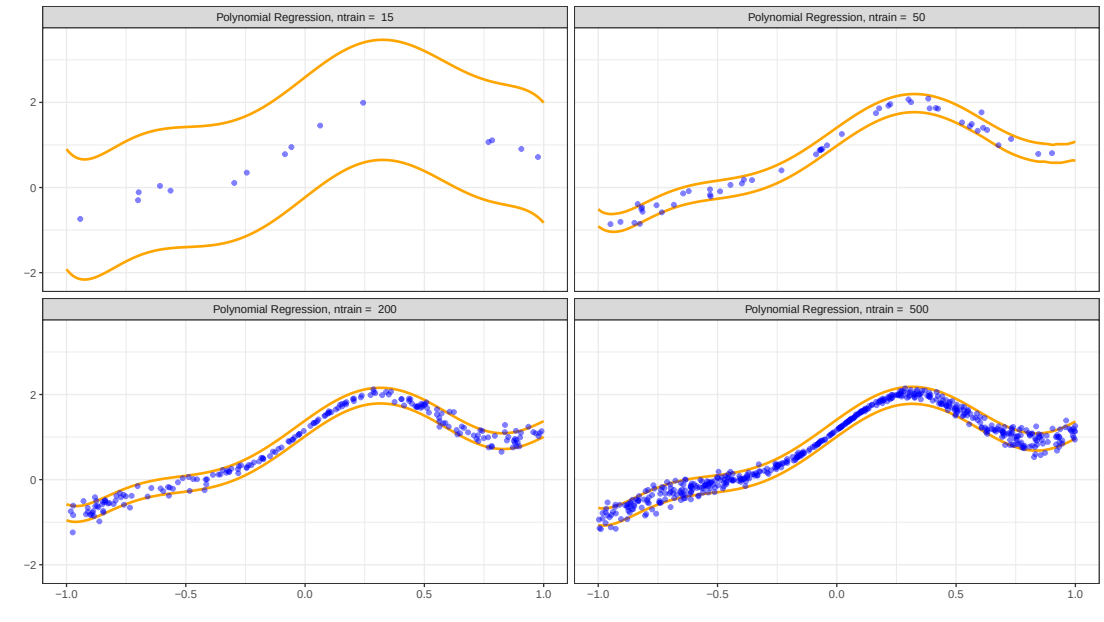
Finally, observe that jackknife+ corresponds to $m = 1$, meaning that without additional assumptions obtaining a training conditional guarantee in this case seems illusory. In fact Bian and Barber (2023) show that it is not possible, both for the jackknife+ and full conformal prediction.

Remark (jackknife+ and CV+ are no longer centered). We will discuss this in the second lecture also, but let us point out that the prediction intervals from jackknife+ and CV+ are no longer centered on a fixed predictor function $\hat{\mu}$, as opposed to split conformal and naive jackknife.

Remark (Computational considerations). This is obvious but should be stated explicitly: you need access to the prediction algorithm and must be able to perform re-training yourself to compute jackknife+ and CV+ intervals, as opposed to split conformal where you just need to be able to run predictions on a calibration set (and not train anything) if someone already gave you a predictor trained on a proper training set.

Example. Still for our one-dimension test case, we take a training dataset $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\ldots,n}$ of increasing size $n = 15, 50, 200, 500$. We only illustrate jackknife+ with polynomial regression, since this is one of the most well-known predictors for which we have very fast

leave-one-out formulas (see the accompanying R code). Training samples from $\mathcal{D}_n$ are represented in blue and jackknife+ conformal intervals for $\alpha = 0.1$ in orange in the figure below.



Once again, we see that the width of the intervals tend to shrink when n increases.

1.4.3 Full conformal prediction

We finish our first overview of traditional conformal prediction methods by the **full conformal prediction** approach, which is sometimes referred to as conformal prediction, without any qualifier at all.

The principle behind full conformal prediction is rather different to what we saw before. One way to introduce it is to first discuss an idealized situation, which cannot happen in practice, where we already know the value of Y_{n+1} (so that we do not need to compute a prediction interval of course). In such a case, we could train a prediction function $\hat{\mu}_{\mathcal{D}_{n+1}}$ on all data $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$, and we know that the event

$$R_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(R_1, \ldots, R_n\right) \tag{1.6}$$

has probability $\geq 1 - \alpha$, where $R_i = |Y_i - \hat{\mu}_{\mathcal{D}_{n+1}}(\boldsymbol{X}_i)|$ are the absolute residuals, by exchangeability and if the prediction algorithm is symmetric. **Be careful** that this is not the same flawed setting as in Equation (1.3) at the beginning of this section, where R_{n+1} was a residual on a test point which was not used in training, now all data are used for training and for computing residuals. Let us then re-write this statement, by clearly separating the effects of $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ and $(\boldsymbol{X}_{n+1}, Y_{n+1})$:

$$|Y_{n+1} - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, Y_{n+1})}(\boldsymbol{X}_{n+1})| \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(|Y_i - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, Y_{n+1})}(\boldsymbol{X}_i)|,\ i = 1, \ldots, n\right)$$

has probability $\geq 1-\alpha$. Now of course we do not know the value of Y_{n+1} (but we know everything else), and building a prediction interval for it just amounts to **finding a set of test values** $y \in \mathcal{Y}$ which may correspond to Y_{n+1} with a certain confidence. Having a look at the previous event, we know that when we use the value Y_{n+1} inside the inequality, it has probability $\geq 1-\alpha$ to be true. Quite naturally, we could then think about testing all values $y \in \mathcal{Y}$ and only retain those for which the inequality is verified. This translates into testing

$$\text{Is } |y - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_{n+1})| \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(|Y_i - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_i)|,\ i = 1, \ldots, n\right)?$$

for all $y \in \mathcal{Y}$. The full conformal interval is finally given by

$$\begin{aligned}
\hat{C}^{\text{full}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) &= \left\{y \in \mathbb{R} :\ |y - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_{n+1})| \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(|Y_i - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_i)|\right)\right\} \\
&= \left\{y \in \mathbb{R} :\ R^y_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(R^y_i,\ i = 1, \ldots, n\right)\right\}
\end{aligned}$$

where $R^y_i = |Y_i - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_i)|$ for $i = 1, \ldots, n$ and $R^y_{n+1} = |y - \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_{n+1})|$. We insist that $\hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}$ corresponds to the prediction function learnt on the augmented dataset $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, y)$, as if y was the true target value observed at $\boldsymbol{X}_{n+1}$. The proof of the following coverage guarantee is then trivial, due to the probability of the event in Equation (1.6) discussed before.

Theorem 5 (Coverage for full conformal - Vovk et al. (2005)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$, $(\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable and the prediction algorithm is symmetric, the full conformal interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{full}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha.$$

Proof. The event $Y_{n+1} \in \hat{C}^{\text{full}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})$ is equivalent to

$$R^{Y_{n+1}}_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(R^{Y_{n+1}}_i,\ i = 1, \ldots, n\right),$$

which is exactly the event in Equation (1.6). □

Remark (Computational considerations). Assembling the full conformal interval is quite tricky because the prediction function must be re-trained:

1. for each test point $\boldsymbol{X}_{n+1}$
2. and for every possible target value $y \in \mathcal{Y}$, with $\mathcal{Y} = \mathbb{R}$ in our case here

Let us comment first the second point because it necessitates special care due to the infinite cardinality of the set of test values. A first useful result states that we can truncate the search to the empirical range of target values in the training data (Chen et al., 2016), and

build ('t' is for *trimmed*):

$$\hat{C}^{\text{full-t}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) = \left\{ y \in [\min Y_i,\ \max Y_i] :\ R^y_{n+1} \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(R^y_i,\ i = 1, \ldots, n\right) \right\}$$

with coverage

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{full-t}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha - \frac{2}{n+1}.$$

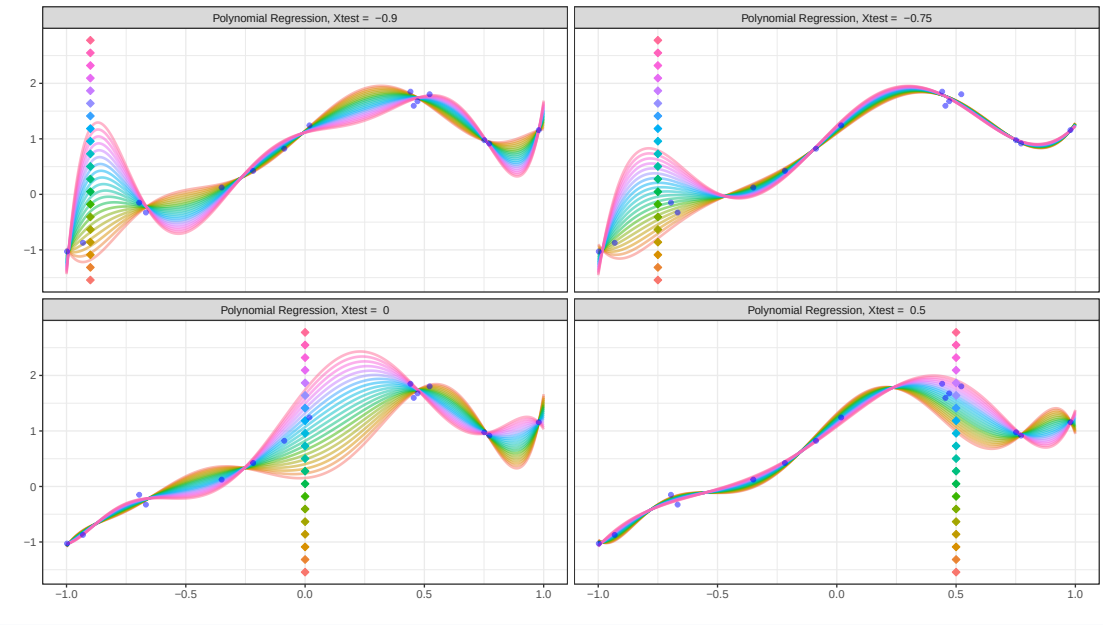
The search space is of course reduced when compared to $\mathcal{Y}$, but it is still infinite and so cannot be implemented in practice. The usual trick is then to use a finite grid of y values and test each of them, but there is no general theory with guaranteed coverage in this case (although this is what is done in many practical implementations).

We can now come back to the first point, because with the discretization trick for y, we will still need to re-train the prediction function for each point $(\boldsymbol{X}_{n+1}, y)$, but now a finite number of times. This will generally be prohibitive, except for specific instances of algorithms called *linear smoothers*. Such supervised learning methods correspond to prediction functions which are linear functions of the training target values $Y_1, \ldots, Y_n$, which implies that the fitted values $\hat{Y}_1, \ldots, \hat{Y}_n$ form a vector writing as $\hat{Y} = SY$ for a $n \times n$ matrix S which depends only on $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. This encompasses linear and ridge regression, nearest neighbors, kernel smoothing, smoothing splines and kernel ridge regression for example (but not random forests nor neural networks).

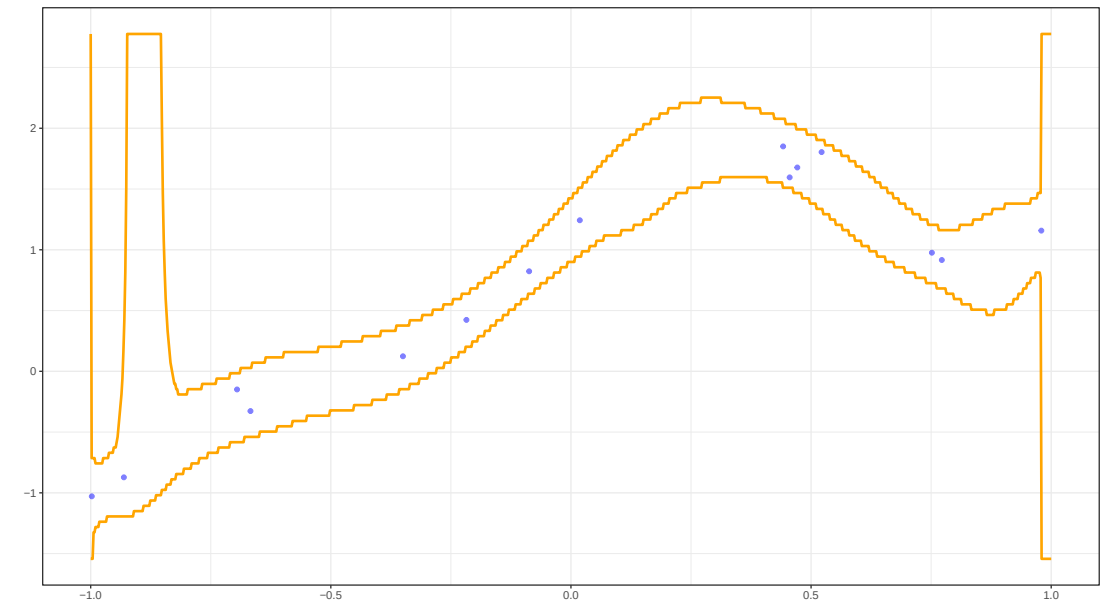
One last comment: similarly to jackknife+ and CV+, for full conformal you also evidently require the capability to re-train the prediction function.

Example. We now only consider a training dataset $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1,\ldots,n}$ of size $n = 15$ and polynomial regression. For full conformal, we use a grid of target values in the range of Y values in the training set (slightly augmented) and use fast formulas using the fact that polynomial regression is a linear smoother (see accompanying R code).

We first show, for several values of $\boldsymbol{X}_{n+1}$, the different predictors obtained when we augment the training data with $(\boldsymbol{X}_{n+1}, y)$ for a grid of y values: the colors below correspond to one value of y (displayed with a colored diamond) and the training samples from $\mathcal{D}_n$ are represented in blue as usual.



Finally, we compute the full conformal intervals with $\alpha = 0.1$ in orange in the figure below for all values of $\boldsymbol{X}_{n+1}$, by identifying the minimum and maximum value of y in the grid that satisfies the full conformal check inequality.



Observe first that the discretization grid is particularly obvious in the plot, as expected. Also, for certain values of $\boldsymbol{X}_{n+1}$ the intervals tend to explode: this corresponds to regions where adding a new training data $(\boldsymbol{X}_{n+1}, y)$ can greatly impact some residuals elsewhere.

Interestingly, the full conformal intervals are highly different from the ones obtained with jackknife+ previously: they are thinner and vary a little more with $\boldsymbol{X}_{n+1}$.

Remark (Split conformal is a (very) special case of full conformal). Let us denote $\mathcal{A}$ the prediction algorithm with outputs the pretrain predictor $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$ when it is given the data $\mathcal{D}_n^{\text{ptrain}}$, and define a new prediction algorithm $\tilde{\mathcal{A}}$ which, when it is given any dataset, will always output $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$ (we say that $\tilde{\mathcal{A}}$ is a non-data-dependent algorithm, and obviously it is symmetric).

Now assuming that $\hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}$ is fixed (we make the dependence to the prediction algorithm explicit now), we can build the full conformal interval with the prediction algorithm $\tilde{\mathcal{A}}$ on the calibration set $\mathcal{D}_n^{\text{cal}}$. In this setting it is given by

$$\begin{aligned}
\hat{C}^{\text{full}}_{\mathcal{D}_n^{\text{cal}}}(\boldsymbol{X}_{n+1}) &= \Big\{ y \in \mathbb{R} : \ |y - \hat{\mu}^{\tilde{\mathcal{A}}}_{\mathcal{D}_n^{\text{cal}} \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_{n+1})| \\
&\quad \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}} \Big(|Y_i - \hat{\mu}^{\tilde{\mathcal{A}}}_{\mathcal{D}_n^{\text{cal}} \cup (\boldsymbol{X}_{n+1}, y)}(\boldsymbol{X}_i)|, \ i \in \mathcal{D}_n^{\text{cal}} \Big) \Big\} \\
&= \Big\{ y \in \mathbb{R} : \ |y - \hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_{n+1})| \\
&\quad \leq \text{Quantile}_{(1-\alpha)\frac{n+1}{n}} \Big(|Y_i - \hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)|, \ i \in \mathcal{D}_n^{\text{cal}} \Big) \Big\} \ \text{(definition of } \tilde{\mathcal{A}} \text{)} \\
&= \hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_{n+1}) \pm \text{Quantile}_{(1-\alpha)\frac{n+1}{n}} \Big(|Y_i - \hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)|, \ i \in \mathcal{D}_n^{\text{cal}} \Big)
\end{aligned}$$

that is the definition of the split conformal interval. The training conditional guarantee for split conformal thus directly derives from the marginal coverage guarantee of $\hat{C}^{\text{full}}_{\mathcal{D}_n^{\text{cal}}}(\boldsymbol{X}_{n+1})$ given $\mathcal{D}_n^{\text{ptrain}}$ (recall that we assumed that $\hat{\mu}^{\mathcal{A}}_{\mathcal{D}_n^{\text{ptrain}}}$ was fixed above).

1.5 Summary and discussion

In this first lecture, we discussed the basic principles behind conformal prediction - finite sample correction of quantiles and exchangeability - and detailed several variants: split conformal, jackknife+ / CV+ conformal and full conformal. In addition, all these methods come with nice theoretical guarantees on the coverage of the intervals they produce.

To go further, let us point out a few facts related to their practical applicability on computer experiment topics:

(i) Training conditional coverage is almost surely a must-have in industry: marginal coverage will not guarantee anything about intervals produced with a fixed training sample (the one that you have at hand in your application) but only on average over all training data you could have randomly generated. Unfortunately only split conformal and CV+

enjoy this property without additional assumptions. *Algorithmic stability* is a key concept which will make it possible to get training conditional coverage for other approaches and will be discussed in the second lecture.

(ii) In illustrations you may have noticed that almost all methods produced intervals with constant width (except full conformal). Intuitively we would have expected something different, with larger width when the predictor does not perform well (for example input regions with few training data) and when there is more noise (remember that our running example has heteroskedastic noise). *Adaptivity* of prediction intervals (in the sense that they vary with $\boldsymbol{x}$) is thus a desirable property, which can be achieved with several methods: this will be investigated in the second lecture as well, with the related concept of *test conditional* coverage.

(iii) Finally, the exchangeability assumption prevents basic conformal prediction from being used in important practical cases, such as time series or active learning (the latter being crucial in computer experiments). We will also present extensions to handle such situations in the next lecture.

Note that along such extensions of conformal prediction, related approaches with the same type of guarantees but different assumptions will also be mentioned.

Lecture 2: Extensions of conformal prediction & related methods

Day 2

- Achieving training conditional coverage
- Distribution shift
- The quest for adaptivity
  - Changing the score function
  - Learning the score function
  - Test conditional coverage
- Concluding remarks

Chapter 2

Extensions of conformal prediction & related methods

Now that we have detailed the most prominent conformal prediction methods with basic exchangeability assumptions, in this second lecture we now discuss several extensions which should be of greater interest from a practical point of view in computer experiments (but not only).

2.1 Achieving training conditional coverage

As elaborated in the first lecture, training conditional guarantee is kind of a must-have in practice. Unfortunately not all conformal methods satisfy this property, and you should be aware of this fact when using conformal prediction on your application.

For illustration purposes, we take inspiration from a numerical experiment in Bian and Barber (2023). We repeat 200 times the following experiment for a fixed training sample size $n = 100$ and a fixed test sample size n_{test}: for a feature dimension $d = n/10, n/2, n-2$, we generate data from a homoskedastic linear model $Y|\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{X}\beta, 1)$ with $\boldsymbol{X} \sim \mathcal{N}(0, I_d)$ and a vector of coefficients β generated at random according to a uniform distribution between $-\sqrt{10}$ and $\sqrt{10}$. For each trial we first compute the jackknife+ prediction intervals for a simple linear regression, and then estimate the conditional coverage by counting the number of test samples that fall in this interval, with $\alpha = 0.1$. We report in Figure 2.1 below the histograms of the training conditional coverage obtained over all repetitions, for each feature dimension d.

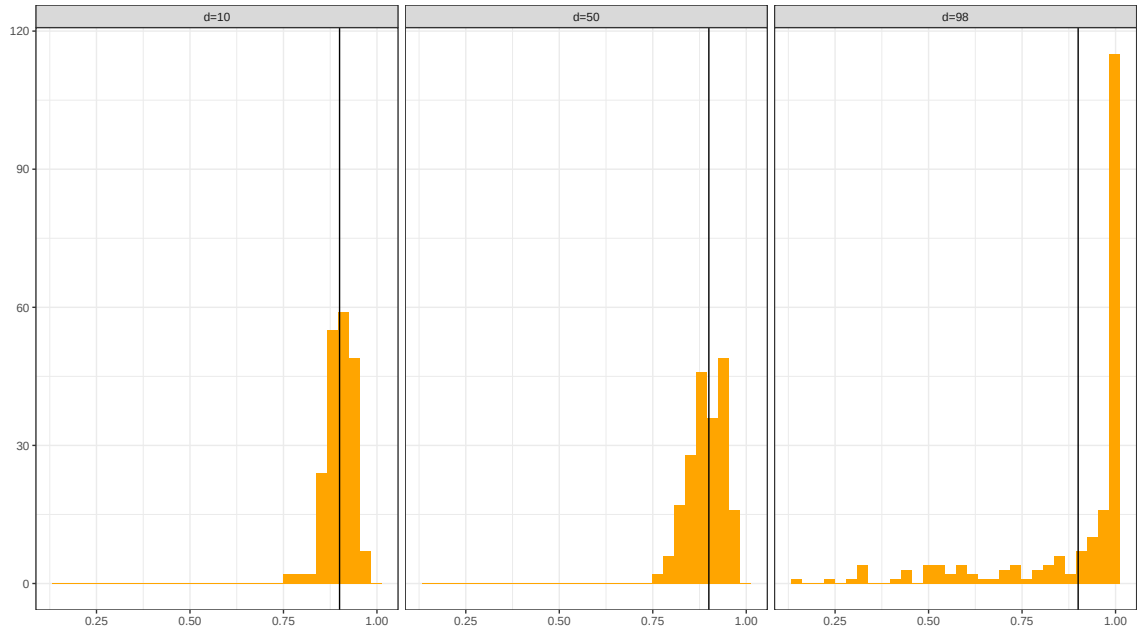


Figure 2.1: Histogram of jackknife+ conditional coverage for linear regression with $n = 100$ and $d = n/10, n/2, n-2$. The target coverage $1-\alpha$ is given as a black vertical line.

Observe first that all histograms are centered around 0.9: this is expected since the marginal coverage, being equal to the mean of the training conditional one, is guaranteed to be around $1-2\alpha$ but close in practice to $1-\alpha$ for the jackknife+. However, a huge difference between the histograms appears when d varies: for $d \ll n$ the fluctuations around $1-\alpha$ are small, but when $d \approx n$ the behavior is quite erratic. For many trials the coverage is much smaller than $1-\alpha$, which is highly problematic, and also for approximately half the repetitions we observe a coverage equal to 1, which occurs because the range of the intervals explode.

In practice we absolutely want to avoid this last situation, because we will be blind: when building intervals with a method with no training conditional guarantee, we may be in a case where our training data give let's say 20% coverage, or in a case where they provide a non-informative interval with 100% coverage.

Fortunately, it is possible to obtain training conditional guarantee for resampling-based conformal methods when the prediction algorithm satisfies a **stability assumption**, that is when the predictions when removing or replacing an observation from the training data are close to each other.

Definition 5 (Algorithmic stability - Barber et al. (2021*b*); Amann et al. (2023); Liang and Barber (2023)). A prediction algorithm $\mathcal{A}$ is (ε_n, ν_n)-stable if the predictor it outputs satisfies $\forall i = 1, \ldots, n$

$$\mathbb{P}\left\{|\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) - \hat{\mu}_{-i}(\boldsymbol{X}_{n+1})| \leq \varepsilon_n\right\} \geq 1 - \nu_n. \quad (\mathcal{SA}_1)$$

A prediction algorithm $\mathcal{A}$ is β_n-stable if the predictor it outputs satisfies $\forall i = 1, \dots, n$

$$\mathbb{E}\left\{|\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) - \hat{\mu}_{-i}(\boldsymbol{X}_{n+1})|\right\} \leq \beta_n. \quad (\mathcal{SA}_2)$$

In both cases the probability is taken over all data $(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$.

Remark ($(\mathcal{SA}_1)$ and $(\mathcal{SA}_2)$ are essentially the same in practice). See Liang and Barber (2023).

In addition to algorithmic stability, many results also involve additional **distributional assumptions**, which help control the conditional distribution $P_{Y|\boldsymbol{X}=\boldsymbol{x}}$. We will come back later in the conclusion and perspectives on how such assumptions may be restrictive for computer experiments.

Assumption 1 (Absolute continuity - $(\mathcal{AC})$). For almost every $\boldsymbol{x}$, the conditional distribution $P_{Y|\boldsymbol{X}=\boldsymbol{x}}$ has Lebesgue density $f_{Y|\boldsymbol{X}=\boldsymbol{x}}$ with finite supremum norm $\|f_{Y|\boldsymbol{X}}\|_\infty = \sup_{y\in\mathbb{R}} f_{Y|\boldsymbol{X}}(y) < \infty$. We also consider assumptions:

- $(\mathcal{AC}^{\mathrm{b}})$ where we further assume that $\|f_{Y|\boldsymbol{X}}\|_\infty$ is bounded by a constant $f_{\sup}$ or
- $(\mathcal{AC}^{\mathrm{e}})$ where we further assume that $\mathbb{E}\|f_{Y|\boldsymbol{X}}\|_\infty$ is bounded by a constant c

Remark (Related stability concepts). In Steinberger and Leeb (2023), the authors define the *K-stability coefficient* η as

$$\eta_n = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left\{\|f_{Y_{n+1}|\boldsymbol{X}_{n+1}}\|_\infty \, |\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) - \hat{\mu}_{-\mathcal{D}_k}(\boldsymbol{X}_{n+1})|\right\}.$$

for a K-fold CV procedure with Assumption $(\mathcal{AC})$. Applied to leave-one-out CV this leads to a stability coefficient equal to

$$\begin{aligned}\eta_n &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left\{\|f_{Y_{n+1}|\boldsymbol{X}_{n+1}}\|_\infty \, |\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) - \hat{\mu}_{-i}(\boldsymbol{X}_{n+1})|\right\} \\ &\leq f_{\sup}\beta_n\end{aligned}$$

where β_n comes from $(\mathcal{SA}_2)$ and we further assume $(\mathcal{AC}^{\mathrm{b}})$. This proves the close links between such stability concepts.

Another related but highly different stability criterion in practice is the *in-sample stability*, as opposed to the *out-sample stability* which is another terminology for the one we defined before. The "in-sample" qualifier refers to the fact that the stability is measured

in average for data used to train both predictors, i.e. the in-sample version of $(\mathcal{SA}_1)$ writes

$$\mathbb{P}\left\{|\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_i) - \hat{\mu}_{-i}(\boldsymbol{X}_i)| \leq \varepsilon_n\right\} \geq 1 - \nu_n. \quad (\mathcal{SA}_1^{\text{in}})$$

This is a much stronger assumption than $(\mathcal{SA}_1)$. Barber et al. (2021*b*) give the example of the k-nearest neighbor predictor, which satisfies $(\mathcal{SA}_1)$ with $\varepsilon_n = 0$, $\nu_n = k/n$ but not $(\mathcal{SA}_1^{\text{in}})$.

Defining the useful concept of stability is great, but knowing which prediction algorithms satisfy such properties is crucial in practice. Here are a few results and references for some of them:

- Linear regression is $(\mathcal{SA})$ and $(\mathcal{SA}^{\text{in}})$ stable unless $d \approx n$ when it is neither. This explains the results we obtained at the very beginning of this section
- Ridge regression (linear or kernel) is $(\mathcal{SA})$ and $(\mathcal{SA}^{\text{in}})$ stable (Bousquet and Elisseeff, 2002), the higher the regularization the more stable
- Lasso regression is $(\mathcal{SA})$ but not $(\mathcal{SA}^{\text{in}})$ stable (Xu et al., 2011)
- Bagging of any base predictor is $(\mathcal{SA})$ (Soloff et al., 2024)
- Random forests are $(\mathcal{SA})$ (Wang et al., 2023)

Now equipped with stability assumptions and control of target conditional distributions, we can first revisit a previous result on marginal coverage before addressing training conditional coverage. These assumptions make it possible to get close to the $1-\alpha$ target, instead of $1-2\alpha$ previously.

Theorem 6 (Marginal coverage $1-\alpha$ for jackknife and jackknife+ - Barber et al. (2021*b*)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable, under Assumptions $(\mathcal{AC}^{\text{e}})$, $(\mathcal{SA}_1)$ and for a symmetric prediction algorithm, then the jackknife interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha - 2\sqrt{\nu_n} - 2\varepsilon_n c$$

and the jackknife+ interval satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack+}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha - 4\sqrt{\nu_n} - 4\varepsilon_n c.$$

Remark. A similar results holds when Assumption $(\mathcal{AC}^{\text{e}})$ is dropped, at the cost of inflating lightly the prediction intervals by a factor of ε_n.

On an interesting side-note, with an additional in-sample stability assumption $(\mathcal{SA}_1^{\text{in}})$,

even the naive prediction interval

$$\hat{C}^{\text{naive}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) \pm \hat{q}^{+}_{n,\alpha}\{|Y_i - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_i)|,\ i = 1, \ldots, n\}$$

discussed in the previous lecture satisfies a similar coverage (Barber et al., 2021*b*)!

We already mentioned that for jackknife and variants, the computational cost may be a limitation. For the specific case of random forests, a recent interesting result shows that intervals built from out-of-bag samples, which do not necessitate re-training, actually achieve marginal coverage guarantees with the same type of assumptions.

Theorem 7 (Marginal coverage for jackknife-OOB on random forests - Wang et al. (2023)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ are i.i.d. and for a random forest satisfying a stability condition $(\mathcal{SA}_1)$, then

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack-oob},\varepsilon_n}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1 - \alpha - \mathcal{O}(\sqrt{\nu_n})$$

where $\hat{C}^{\text{jack-oob},\varepsilon_n}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}^{\text{RF}}_{\mathcal{D}_n}(\boldsymbol{x}) \pm \hat{q}^{+}_{n,\alpha}\{|Y_i - \hat{\mu}^{\text{RF}}_{\text{oob},-i}(\boldsymbol{X}_i)| + \varepsilon_n,\ i = 1, \ldots, n\}$ with $\hat{\mu}^{\text{RF}}_{\mathcal{D}_n}$ the random forest predictor and $\hat{\mu}^{\text{RF}}_{\text{oob},-i}$ the out-of-bag predictor for observation i.

Now, the main results for this section concern training conditional guarantees for the jackknife and the jackknife+, to complete the picture (full conformal is discussed in the remark below). Note that we provide here **asymptotic guarantees**. Although finite-sample results exist, they are not very useful in practice because they involve unknown constants (in general), similarly to the marginal coverages we just detailed. As such, you should remember that in the majority of cases, training conditional coverage will unfortunately not have the desirable finite-sample property, and neither the distribution-free one since we usually require some $(\mathcal{AC})$-like assumption: this is a reason why I tend to say that conformal prediction is not magical and is not as useful as advertised in a distribution-free and finite-sample setting (this is only my personal opinion). On the contrary, stability assumptions hold for many predictors you would use in practice.

Theorem 8 (Asymptotic training conditional coverage for jackknife+ - Liang and Barber (2023)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ are exchangeable, under Assumptions $(\mathcal{AC}^{\text{e}})$, $(\mathcal{SA}_2)$ and for a symmetric prediction algorithm, if $\beta_n \to 0$ then the jackknife+ interval satisfies

$$\lim_{n\to\infty} \mathbb{P}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack}+}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n\right) \geq 1 - \alpha - \delta_n\right\} \to 1$$

for some sequence $\delta_n \to 0$.

Theorem 9 (Asymptotic training conditional coverage for jackknife - Steinberger and Leeb (2023); Amann et al. (2023)). If $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n), (\boldsymbol{X}_{n+1}, Y_{n+1})$ are i.i.d., under As-

sumptions ($\mathcal{AC}^{\text{b}}$), ($\mathcal{SA}_2$) and a boundedness assumption on the prediction error, if $\eta_n \to 0$ then the jackknife interval satisfies

$$\lim_{n\to\infty} \mathbb{E}\left\{\left|\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{jack}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})|\mathcal{D}_n\right) - (1-\alpha)\right|\right\} \to 0.$$

Remark. Both results are highly similar, and only differ in their assumptions: exchangeability and algorithm symmetry for the former (with a small overshooting δ_n of the coverage level), and bounded prediction error and i.i.d. samples for the latter.

Note that Liang and Barber (2023) also show training conditional coverage for the full conformal intervals with an in-sample stability assumption ($\mathcal{SA}_2^{\text{in}}$), the in-sample version of Assumption ($\mathcal{SA}_2$).

Finally, Steinberger and Leeb (2023) prove the statement for K-fold CV and the jackknife result is just a particular case.

2.2 Distribution shift

Distribution shift refers to situations where either the training data $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ and the test data $(\boldsymbol{X}_{n+1}, Y_{n+1})$ do not have the same underlying generating probability distribution, or more generally if all $(\boldsymbol{X}_i, Y_i)$ do not have the same distribution. Obviously this means that data are not i.i.d., but also that there is no hope to have exchangeability either, thus rendering all previous results useless in this setting.

Addressing this problem in all generality is a hard task, this is why we focus here on one specific instance that is of particular interest in practice: the *covariate shift* setting. In this case we assume that the marginal distribution of the features (or covariates) $\boldsymbol{X}$ differs in training and test data, but the conditional distribution of $Y|\boldsymbol{X}$ is the same. In other words, we have $P^{\text{train}}_{\boldsymbol{X}Y} = P^{\text{train}}_{\boldsymbol{X}} \times P_{Y|\boldsymbol{X}}$ and $P^{\text{test}}_{\boldsymbol{X}Y} = P^{\text{test}}_{\boldsymbol{X}} \times P_{Y|\boldsymbol{X}}$. Interestingly, in this case, it is possible to perform **weighted conformal prediction** by using quantiles on a weighted empirical distribution of the data and still guarantee marginal coverage. This is formalized in the following theorem.

Theorem 10 (Marginal coverage for weighted full conformal intervals - Tibshirani et al. (2019)). Assume we are in the covariate shift setting above with $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ i.i.d. from $P^{\text{train}}_{\boldsymbol{X}Y}$ and $(\boldsymbol{X}_{n+1}, Y_{n+1})$ from $P^{\text{test}}_{\boldsymbol{X}Y}$ independently. Define

$$\hat{q}^{w}_{n,\alpha}\{v_i \cup \{+\infty\}\} = \text{Quantile}_{1-\alpha}\left(\frac{1}{\sum_j w(\boldsymbol{X}_j)}\sum_{i=1}^{n} w(\boldsymbol{X}_i)\delta_{v_i} + \frac{w(\boldsymbol{X}_{n+1})}{\sum_j w(\boldsymbol{X}_j)}\delta_{+\infty}\right).$$

Then the full conformal interval

$$\hat{C}^{\text{full}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) = \left\{y \in \mathbb{R} : \; R^y_{n+1} \leq \hat{q}^w_{n,\alpha}\{(R^y_i)_{i=1,\dots,n} \cup \{+\infty\}\}\right\}$$

satisfies

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{full}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})\right) \geq 1-\alpha$$

for the weight function $w = \mathrm{d}P^{\text{test}}_{\boldsymbol{X}}/\mathrm{d}P^{\text{train}}_{\boldsymbol{X}}$ where we further assume that $P^{\text{test}}_{\boldsymbol{X}}$ is absolutely continuous with respect to $\mathrm{d}P^{\text{train}}_{\boldsymbol{X}}$.

Remark. Observe first that if there is no covariate shift, the weight function is the identity and the weighted quantile becomes

$$\begin{aligned}\hat{q}^w_{n,\alpha}\{v_i \cup \{+\infty\}\} &= \text{Quantile}_{1-\alpha}\left(\frac{1}{n+1}\sum_{i=1}^n \delta_{v_i} + \frac{1}{n+1}\delta_{+\infty}\right) \\ &= \text{Quantile}_{(1-\alpha)\frac{n+1}{n}}\left(v_i\right)\end{aligned}$$

for $\alpha \geq 1/(n+1)$ and we recover the usual full conformal interval.

Another important point is that the weight function, which is a likelihood ratio, must be known in advance. However for the particular case where the probability density functions in the training and test set are known only up to a constant, the result still applies because the constants cancel out in the weighted empirical distribution function.

Finally, recall that in the first lecture we showed that split conformal can be seen as a particular case of full conformal. This implies that weighted split conformal intervals also have such coverage guarantees (more precisely the training conditional one, as usual with split conformal), i.e. the interval

$$\hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}^{\text{ptrain}}_n}(\boldsymbol{x}) \pm \hat{q}^w_{n,\alpha}\{(R_i)_{i\in\mathcal{D}^{\text{cal}}_n} \cup \{+\infty\}\}$$

satisfies the training conditional guarantee

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}^{\text{ptrain}}_n\right) \geq 1-\alpha$$

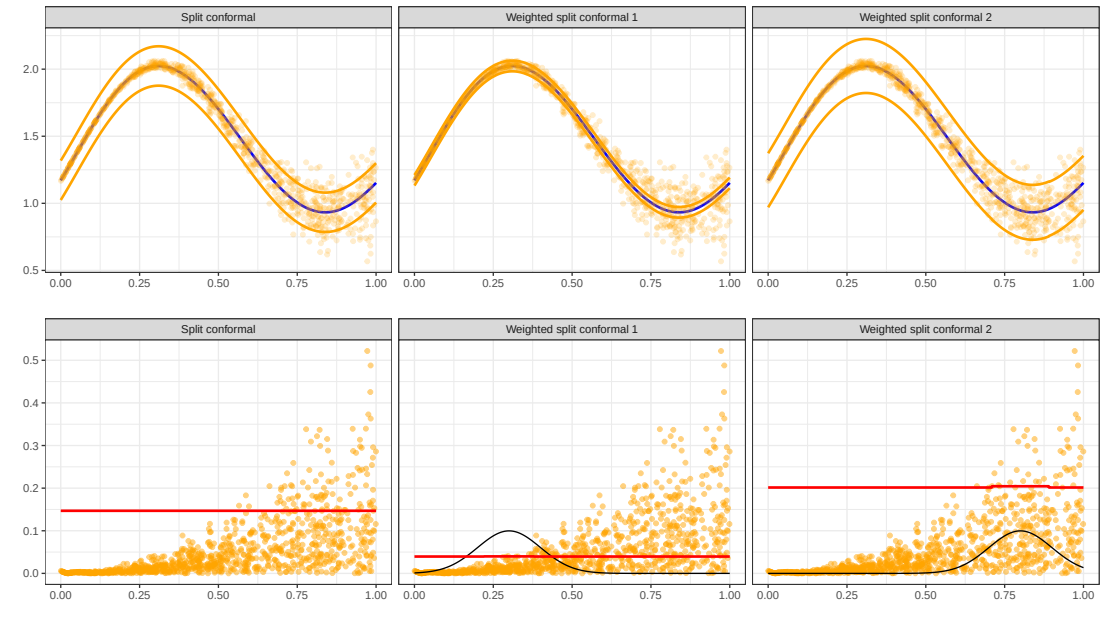
for the same weight functions as in the previous theorem, see supplementary material of Tibshirani et al. (2019).

Example. Let us illustrate weighted split conformal intervals on a slight variation of our running example:

$$Y = X^3 + 2\exp(-6(X-0.3)^2) + \varepsilon$$

where $\varepsilon \sim \mathcal{N}(0, 0.2|X|)$ and $X \sim P^{\text{train}}_{\boldsymbol{X}} = \mathcal{U}[0,1]$ for the training set (of size $n = 2000$) and

we consider two covariate shift settings where $X \sim P_{\boldsymbol{X}}^{\text{test},1} = \mathcal{N}(0.3, 0.1)$ and $X \sim P_{\boldsymbol{X}}^{\text{test},2} = \mathcal{N}(0.8, 0.1)$ for the test set. We run split conformal prediction with $n_{\text{cal}} = 1000$ and $\alpha = 0.1$ with the original unweighted quantile, and the weighted quantiles for $P_{\boldsymbol{X}}^{\text{test},1}$ and $P_{\boldsymbol{X}}^{\text{test},2}$. We show below the prediction (blue line), the prediction intervals (orange line) and calibration data (orange dots) for each method in the top row. The bottom row displays, for each method, the calibration data residuals (orange dots), the quantile value (red line) and the distribution of the shifted covariate if any (black line).



The impact of the weighted quantiles is particularly clear:

- $P_{\boldsymbol{X}}^{\text{test},1}$ concentrates on regions where the original split conformal has overcoverage, so that the weighted quantile is smaller, resulting in narrower prediction intervals in the end
- At the opposite, $P_{\boldsymbol{X}}^{\text{test},2}$ corresponds to regions where we have undercoverage for unweighted split conformal, the weighted quantile is thus larger and the weighted prediction intervals are larger

We can quantitatively assess these findings by computing the training conditional coverage of each method with respect to both $P_{\boldsymbol{X}}^{\text{test},1}$ and $P_{\boldsymbol{X}}^{\text{test},2}$, see the results in the table below.

	Split conformal	Weighted split conformal 1	Weighted split conformal 2
Coverage $P_{\boldsymbol{X}}^{\text{test},1}$	1	**0.91**	1
Coverage $P_{\boldsymbol{X}}^{\text{test},2}$	0.735	0.247	**0.863**

Remark (Weighted split intervals are typically constant). We have seen that by design, split conformal produces prediction intervals with constant width. Looking closely at the definition of the weighted quantiles, we can see that this time they depend on the value of the test point $\boldsymbol{X}_{n+1}$, meaning that theoretically they are not constant. However, when n_{cal} is sufficiently large, the impact of $\boldsymbol{X}_{n+1}$ is negligible in the computation, thus yielding also constant intervals: this can be observed in the example above. The same comment applies to weighted full conformal prediction of course.

For weighted conformal prediction, we emphasized that the shifted distribution of the covariates $P_{\boldsymbol{X}}^{\text{test}}$ must be known in advance. Although this is the case in some examples, a more representative setting of practical applications would be that, if we consider $P_{\boldsymbol{X}}^{\text{test}}$ as a region of interest where we want to guarantee coverage, such a region will most often come from insights gained from the trained predictor itself. This could be feature regions where the target is predicted to be close to a threshold (for reliability studies) or where it is small (for minimization problems), among others. In this setting, this means that $P_{\boldsymbol{X}}^{\text{test}}$ is not given beforehand, but is deduced from the training data (through the predictor), and we can formally write

$$P_{\boldsymbol{X}Y}^{\text{train}} = P_{\boldsymbol{X}}^{\text{train}} \times P_{Y|\boldsymbol{X}}, \quad P_{\boldsymbol{X}Y}^{\text{test}} = P_{\boldsymbol{X};\mathcal{D}_n}^{\text{test}} \times P_{Y|\boldsymbol{X}}$$

where we made this dependence explicit. Theoretically such assumptions no longer fall in the weighted conformal prediction framework, but recently it has been shown that actually we can still have coverage in this generalized setting, as formalized in the following theorem.

Theorem 11 (Training conditional coverage for weighted split conformal intervals based on training data - Fannjiang et al. (2022)). Assume we are in the covariate shift setting above with $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ i.i.d. from $P_{\boldsymbol{X}Y}^{\text{train}}$ which is split in $\mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{cal}}$ and denote $P_{\boldsymbol{X};\mathcal{D}_n^{\text{ptrain}}} = P_{\boldsymbol{X};\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}}$ a shifted covariate distribution which depends on $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$, a predictor trained on $\mathcal{D}_n^{\text{ptrain}}$. For the residuals on the calibration data $R_i = |Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)|$, $i \in \mathcal{D}_n^{\text{cal}}$ we consider the weighted split prediction interval

$$\hat{C}_{\mathcal{D}_n}^{\text{split}}(\boldsymbol{X}_{n+1}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}} \pm \hat{q}_{n,\alpha}^{w}\{(R_i)_{i\in\mathcal{D}_n^{\text{cal}}} \cup \{+\infty\}\}$$

with the weight function given by $w = \mathrm{d}P_{\boldsymbol{X};\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}} / \mathrm{d}P_{\boldsymbol{X}}^{\text{train}}$. Then, if $\mathrm{d}P_{\boldsymbol{X};\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}}$ is absolutely continuous with respect to $\mathrm{d}P_{\boldsymbol{X}}^{\text{train}}$, we have

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}^{\text{split}}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{ptrain}}\right) \geq 1 - \alpha$$

for $(\boldsymbol{X}_{n+1}, Y_{n+1})$ from $P_{\boldsymbol{X}Y}^{\text{test}} = P_{\boldsymbol{X};\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}} \times P_{Y|\boldsymbol{X}}$ independently. Marginal coverage is obtained if $P_{\boldsymbol{X};\mathcal{D}_n^{\text{ptrain}}}$ is absolutely continuous with respect to $\mathrm{d}P_{\boldsymbol{X}}^{\text{train}}$ for all pretraining datasets $\mathcal{D}_n^{\text{ptrain}}$.

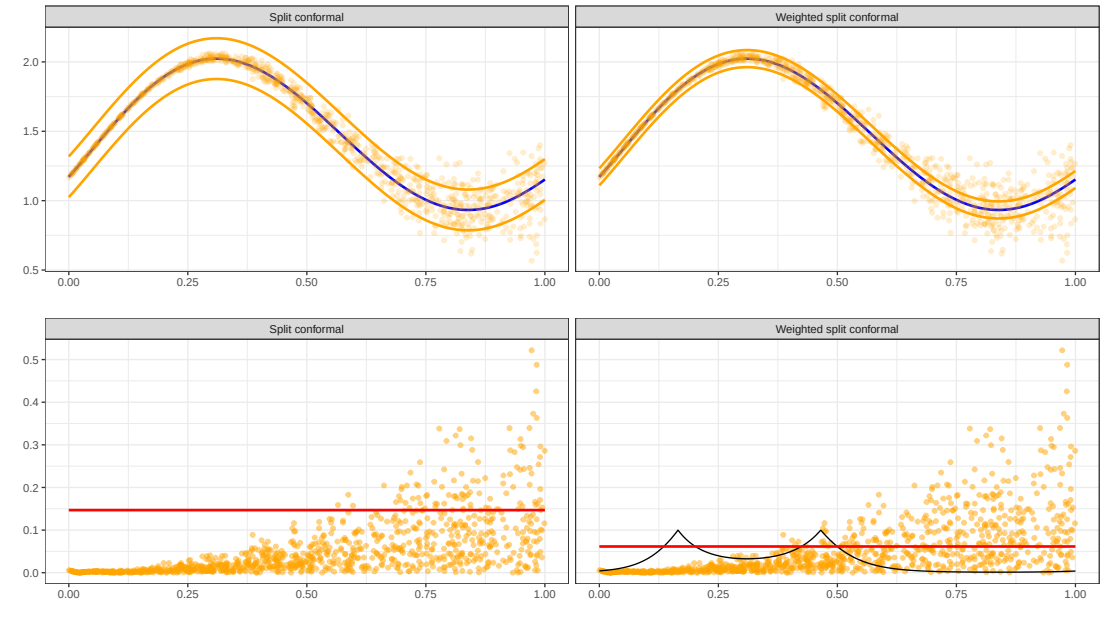
Remark. In Fannjiang et al. (2022) the main result actually concerns full conformal prediction, but the notations are more complicated (since the weights depend on re-training steps of the predictor on virtual points). This explains why we only state the split conformal version which is found in Appendix A1.3 of Fannjiang et al. (2022).

Such a result is very useful and remarkable for computer experiments, since it opens the path for coverage guarantees for *active learning* methods. For optimization specifically, recent ideas close to this principle have been proposed for conformalized Bayesian optimization (Stanton et al., 2023; Deshpande et al., 2024).

Example. For our modified running example above, imagine that we want to have coverage guarantees when the target is close to a value of $t = 1.8$, which could be seen as some kind of critical region for our application. We thus consider a shifted distribution given by

$$P_{\boldsymbol{X};\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}}(\boldsymbol{x}) \propto \exp\left(-\lambda|\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) - t|\right)$$

where $\lambda > 0$ allows to focus more on less on the region where we are close of the threshold t. We run the exact same experiment as before with this new distribution, with $\lambda = 5$ below.



Notice how the shifted distribution automatically concentrates in the two regions where the predictor is close to the threshold.

Appart from this specific case of covariate shift, the more general problem of *distribution drift* where the distribution of training data also evolves (with time for example for timeseries)

is more tricky, even if recent attempts were made towards general coverage guarantees, see for example Barber et al. (2023); Zaffran et al. (2022).

2.3 The quest for adaptivity

Another key aspect of the conformal methods discussed so far is that in general they will tend to produce intervals with constant width (i.e. which does not depend on $\boldsymbol{x}$):

- $\hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \hat{q}^+_{n,\alpha}\left\{R_i,\ i \in \mathcal{D}_n^{\text{cal}}\right\}$ from split conformal has constant width by definition
- The same is true for $\hat{C}^{\text{jack}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x}) \pm \hat{q}^+_{n,\alpha}\{R_i,\ i = 1, \ldots, n\}$ from jackknife
- This is approximately true for $\hat{C}^{\text{jack}+}_{\mathcal{D}_n}(\boldsymbol{x}) = \left[\hat{q}^-_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) - R_i^{\text{LOO}}\}, \hat{q}^+_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) + R_i^{\text{LOO}}\}\right]$, since for predictors with algorithmic stability we know that $\hat{\mu}_{-i}(\boldsymbol{x})$ will be close to $\hat{\mu}_{\mathcal{D}_n}(\boldsymbol{x})$, and most of the predictors you use in practice are stable (recall the list in the previous section)
- However this may not be the case for full conformal (as we observed during the first lecture)

This feature is not desirable in applications, since we expect by intuition that the intervals should be wider in regions where we have less training data, or when the noise is higher if we are in an heteroskedastic situation. In other words, we look for intervals with the property of *adaptivity*, sometimes referred to as *local adaptivity*. We will see in this section how we can address this limitation through the concept of *score function*. At this point let us mention that we did not formalize what adaptivity means from a theoretical point of view, this will be touched upon at the end of the section with the notion of *test conditional coverage*.

2.3.1 Score function

Up until now, the design of prediction intervals relied on the absolute residuals $|Y - \hat{\mu}(\boldsymbol{X})|$ as a measure of the uncertainty of $\hat{\mu}$ for predicting the target, the lower the better. This is thus a *score* attributed to the predictor. Interestingly, since we mainly used exchangeability arguments before to prove the theorems, it happens that we are not restricted to such residuals.

Changing the score function Mathematically, a **score function** $S : (\mathcal{X}\times\mathcal{Y}) \times (\mathcal{X}\times\mathcal{Y})^n \to \mathbb{R}$ maps a data point from $(\mathcal{X} \times \mathcal{Y})$ and a data set of size n from $(\mathcal{X} \times \mathcal{Y})^n$ to a real number indicating if the data point is typical from the data set. For example, we can set

$$S((\boldsymbol{X}, Y), \mathcal{D}_n) = |Y - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X})|$$

where $\hat{\mu}_{\mathcal{D}_n}$ is a predictor trained on $\mathcal{D}_n$, and S is the absolute residual considered before, with a large value indicating that the predictor does not predict well the point $(\boldsymbol{X}, Y)$, and thus

that it may be atypical of the joint distribution observed in the training set. By a slight abuse of notation, we will also sometimes consider a score function written as $S((\boldsymbol{X}, Y), \hat{\mu}_{\mathcal{D}_n})$ where the dependence on the data is only through a trained predictor $\hat{\mu}_{\mathcal{D}_n}$.

Focusing now on the theoretical results we discussed before, we actually can choose any score function:

- For split conformal, instead of the residuals $R_i = |Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)|$, $i \in \mathcal{D}_n^{\text{cal}}$ for a predictor trained on $\mathcal{D}_n^{\text{ptrain}}$, define $S_i = S\left((\boldsymbol{X}_i, Y_i), \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}\right)$ for any score function S. Conditionally on $\mathcal{D}_n^{\text{ptrain}}$, $(S_i)_{i \in \mathcal{D}_n^{\text{cal}}}, S_{n+1}$ are exchangeable by Proposition 1 (iv), and thus a prediction interval of the form

$$\hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{x}) = \left\{ y \in \mathbb{R} : \; S\left((\boldsymbol{x}, y), \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}\right) \leq \hat{q}^{+}_{n,\alpha}\left\{S_i, \; i \in \mathcal{D}_n^{\text{cal}}\right\} \right\}$$

  satisfies the training conditional guarantee

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{ptrain}}\right) \geq 1 - \alpha$$

  because

$$\begin{aligned} Y_{n+1} \in \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) \quad &\Leftrightarrow \quad S\left((\boldsymbol{X}_{n+1}, Y_{n+1}), \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}\right) \leq \hat{q}^{+}_{n,\alpha}\left\{S_i, \; i \in \mathcal{D}_n^{\text{cal}}\right\} \\ &\Leftrightarrow \quad S_{n+1} \leq \hat{q}^{+}_{n,\alpha}\left\{S_i, \; i \in \mathcal{D}_n^{\text{cal}}\right\} \end{aligned}$$

  and the result follows from exchangeability.

- Similarly, we do not detail the proof here, but for full conformal, any score function $S((\boldsymbol{X}, Y), \mathcal{D}_n)$ which is symmetric with respect to the data points in $\mathcal{D}_n$ will work (since we need to suppose the algorithm is symmetric for this method). This means we will consider, instead of R_i^y before:

$$\begin{aligned} S_i^y &= S\left((\boldsymbol{X}_i, Y_i), \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}\right) \; i = 1, \ldots, n \\ S_{n+1}^y &= S\left((\boldsymbol{X}_{n+1}, y), \hat{\mu}_{\mathcal{D}_n \cup (\boldsymbol{X}_{n+1}, y)}\right) \end{aligned}$$

- For jackknife+ and CV+ conformal, I am not aware of results for general score functions, but only for some specific forms, one of them being discussed below. But before CV+ was introduced, a related method called **cross-conformal** prediction was proposed with any symmetric score, with an interval given by

$$\hat{C}^{\text{cross}}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) = \left\{ y \in \mathbb{R} : \; \sum_{k=1}^{K} \sum_{i \in \mathcal{D}_k} \mathbb{1}_{S\left((\boldsymbol{X}_{n+1}, y), \hat{\mu}_{-\mathcal{D}_k}\right) \leq S\left((\boldsymbol{X}_i, Y_i), \hat{\mu}_{-\mathcal{D}_k}\right)} \geq \alpha(n+1) \right\},$$

  see Vovk et al. (2018), with coverage equal to the CV+ case (Barber et al., 2021*b*).

Now, since we are not restricted by the choice of a score function, which form should we choose to achieve local adaptivity? The most popular type is a simple rescaled version of the absolute residuals:

$$S((\boldsymbol{X}, Y), \mathcal{D}_n) = \frac{|Y - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X})|}{\hat{\sigma}_{\mathcal{D}_n}(\boldsymbol{X})} \tag{2.1}$$

where $\hat{\sigma}_{\mathcal{D}_n}$ is trained on $\mathcal{D}_n$, as an estimator of the conditional standard deviation the same way $\hat{\mu}_{\mathcal{D}_n}$ is trained to estimate the conditional mean in general, see for example Lei et al. (2018). In practice, $\hat{\sigma}_{\mathcal{D}_n}$ can be any such estimator obtained from $\hat{\mu}_{\mathcal{D}_n}$: if you recall our very first example in the first lecture, you may think about asymptotic results, bayesian approaches or heuristics based on ensembles, among others. But it can also be a much simpler approach, such as computing a new predictor on the dataset $(\boldsymbol{X}_i, |Y_i - \hat{\mu}_{\mathcal{D}_n}(\boldsymbol{X}_i)|)_{i=1,\dots,n}$ after having trained $\hat{\mu}_{\mathcal{D}_n}$ (Lei et al., 2018). Interestingly, for this specific form of score function, the obtained split conformal intervals are explicit:

$$\begin{aligned} \hat{C}^{\text{split}}_{\mathcal{D}_n}(\boldsymbol{x}) &= \left\{ y \in \mathbb{R} : \; S\left((\boldsymbol{x}, y), \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}\right) \leq \hat{q}^{+}_{n,\alpha}\left\{S_i, \; i \in \mathcal{D}_n^{\text{cal}}\right\}\right\} \\ &= \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \hat{\sigma}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \hat{q}^{+}_{n,\alpha}\left\{S_i, \; i \in \mathcal{D}_n^{\text{cal}}\right\}. \end{aligned}$$

Moreover, this form also makes it possible to generalize jackknife+ and CV+ intervals in this specific instance, by considering

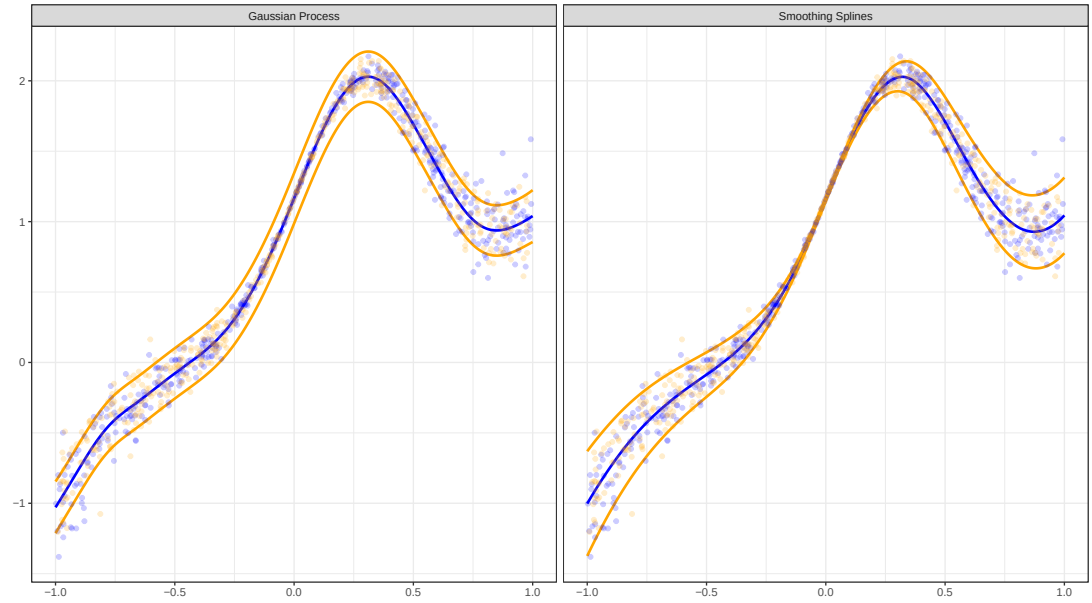
$$\hat{C}^{\text{jack+}}_{\mathcal{D}_n}(\boldsymbol{x}) = \left[\hat{q}^{-}_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) - \hat{\sigma}_{-i}(\boldsymbol{x}) S_i^{\text{LOO}}\}, \hat{q}^{+}_{n,\alpha}\{\hat{\mu}_{-i}(\boldsymbol{x}) + \hat{\sigma}_{-i}(\boldsymbol{x}) S_i^{\text{LOO}}\}\right]$$

where $S_i^{\text{LOO}} = |Y_i - \hat{\mu}_{-i}(\boldsymbol{X}_i)|/\hat{\sigma}_{-i}(\boldsymbol{X}_i)$, see for example a proof in Jaber et al. (2024).

Example. On our running example, we illustrate split conformal prediction with the rescaled score in Equation (2.1) obtained with two methods:

- A GP predictor for $\hat{\mu}$ and its associated posterior variance for $\hat{\sigma}$
- A smoothing spline for $\hat{\mu}$, and for $\hat{\sigma}$ a second smoothing spline trained on the absolute residuals of the first one

We take a training set $\mathcal{D}_n$ of size $n = 1000$, a calibration set of size $n_{\text{cal}} = 500$ and $\alpha = 0.1$. In the figure below, pretraining data are represented with blue dots, calibration data with orange dots, the predictor with a blue line and prediction intervals with orange lines.



Remarkably, the prediction intervals obtained with rescaled scores are now much more adaptive, in particular in the middle where there is much less noise.

For adaptivity, another line of work initiated by Romano et al. (2019) kind of changes the paradigm of conformal prediction which often focuses on residuals (weighted or not), and comes back to the obvious: the oracle prediction interval would be obtained by the quantiles of the true conditional distribution function of the target given the features. Consequently, instead of considering a predictor trained to estimate the conditional mean (and the conditional variance for rescaled scores) and somehow work around the corresponding residuals to obtain valid intervals, why not focus directly in estimating the conditional quantiles and then post-process them to get valid intervals?

This is exactly the principle behind **conformalized quantile regression** (Romano et al., 2019), which is similar to split conformal prediction but starts by building two predictors on the pretraining set, $\hat{\mu}^{\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}$ and $\hat{\mu}^{1-\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}$, which are estimators of the conditional quantiles of level $\alpha/2$ and $1-\alpha/2$. Such predictors are obtained with quantile regression supervised learning methods, hence the name. From them we can build scores defined as

$$S\left((\boldsymbol{X}, Y), \mathcal{D}_n^{\text{ptrain}}\right) = \max\left(\hat{\mu}^{\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}) - Y, Y - \hat{\mu}^{1-\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X})\right)$$

which are then computed on the calibration set to finally produce prediction intervals writing

$$\hat{C}^{\text{CQR}}_{\mathcal{D}_n}(\boldsymbol{x}) = \left[\hat{\mu}^{\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) - \hat{q}^{+}_{n,\alpha}\left\{S_i,\ i \in \mathcal{D}_n^{\text{cal}}\right\}, \hat{\mu}^{1-\alpha/2}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) + \hat{q}^{+}_{n,\alpha}\left\{S_i,\ i \in \mathcal{D}_n^{\text{cal}}\right\}\right]$$

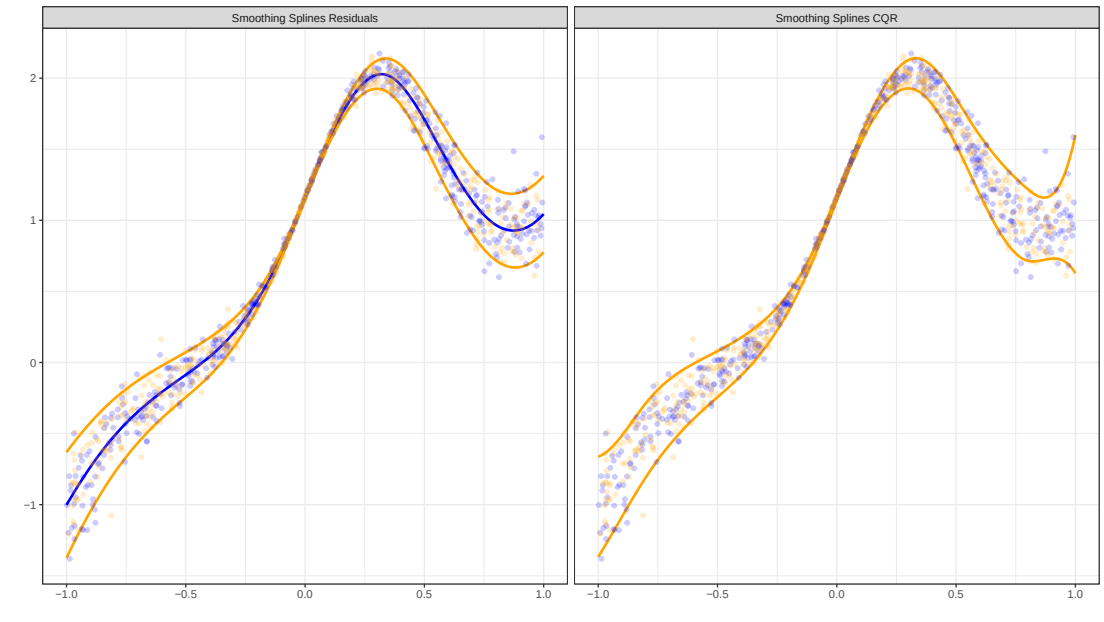
with guaranteed $1-\alpha$ coverage.

Related ideas based on estimation of the conditional distribution function (Izbicki et al., 2019; Chernozhukov et al., 2021) or the conditional density function (Izbicki et al., 2019) have also been introduced recently.

Remark. When focusing on conditional quantile estimation instead of the traditional conditional mean, it is worthy to note that:

- The prediction intervals will obviously not be centered around a traditional predictor of the conditional mean
- You need to have access to implementations allowing quantile regression (theoretically we could say that it is as simple as just changing a squared loss with a pinball loss, but in practice this is more complicated than just that)
- It is expected to yield more adaptive intervals if the noise is not symmetric when compared to rescaled residuals

Example. In the exact same setting as before, we also compute the CQR prediction intervals where quantile regression is performed with smoothing splines, see the figure below. Except at the right endpoint, both approaches here are highly similar.



Learning the score function We just saw that local adaptivity can be greatly improved if we change the score function, and make it depend on estimates of features of the conditional distribution (standard deviation or quantiles). But intrinsically, such features are learnt in some kind of black-box and unsupervised way, in the sense that they are functions trained on the data without accounting for the main target, that is coverage. In a complementary line of work, some recent proposals suggested to instead learn directly functions involved in the prediction intervals, with a constraint on the coverage during training (Liang, 2022; Fan et al., 2023).

For illustration, let us focus on a symmetric prediction interval centered on a predictor $\hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}$ trained on a proper training set $\mathcal{D}_n^{\text{ptrain}}$ and we have an independent set $\mathcal{D}_n^{\text{opt}}$. The idea is to use these data to learn a function $\hat{f}_{\mathcal{D}_n^{\text{opt}}}$ such that the prediction interval

$$\hat{C}^{100\%}_{\mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{opt}}}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \sqrt{\hat{f}_{\mathcal{D}_n^{\text{opt}}}(\boldsymbol{x}) + \delta}$$

has 100% coverage on the optimization set $\mathcal{D}_n^{\text{opt}}$, for $\delta > 0$ a small constant decreasing to 0 with n. In words, $\hat{f}_{\mathcal{D}_n^{\text{opt}}}$ will serve in this setting as an estimator of the conditional variance, but trained with coverage constraints and potential regularization. In a last step, another independent data set $\mathcal{D}_n^{\text{cal}}$ is used to adjust this preliminary interval to the target level $1-\alpha$ just as in split conformal, and build the final prediction interval

$$\hat{C}^{\text{univ}}_{\mathcal{D}_n}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \sqrt{\hat{\lambda}^{\alpha}_{\mathcal{D}_n^{\text{cal}}}(\hat{f}_{\mathcal{D}_n^{\text{opt}}}(\boldsymbol{x}) + \delta)}$$

by identifying a suitable factor $\hat{\lambda}^{\alpha}_{\mathcal{D}_n^{\text{cal}}}$ ("univ" here stands for universal following the claims by Liang (2022); Fan et al. (2023)). Remark that now we denote $\mathcal{D}_n = \mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{opt}} \cup \mathcal{D}_n^{\text{cal}}$.

The interesting point is how we can build the preliminary interval $\hat{C}^{100\%}_{\mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{opt}}}(\boldsymbol{x})$ and write a supervised learning problem with standard tools to do so. The unknown is an infinite dimensional function $\hat{f}$: we know that we will have to impose constraints to guarantee 100% coverage, but we may also want to impose regularity or some other features. We also need to specify $\mathcal{F}$, a space of candidate functions (hypothesis set). All in one, we may write $\hat{f}$ as the solution of a general optimization problem given by

$$\begin{aligned} \min_{f \in \mathcal{F}} \quad & L\left(f, (\boldsymbol{X}_i, Y_i)_{i \in \mathcal{D}_n^{\text{opt}}}\right) \\ \text{s.t.} \quad & f(\boldsymbol{X}_i) \geq \left(Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)\right)^2, \ \forall i \in \mathcal{D}_n^{\text{opt}}. \end{aligned} \tag{2.2}$$

Note that in this problem:

- $\mathcal{F}$ can be chosen as anyone wishes, the complexity of the corresponding class of functions will however appear in the theoretical results
- L denotes a general loss function which encompasses any property or regularization penalty we may want to impose on f, we will give examples below
- The constraints aim at guaranteeing that the coverage attains 100% with respect to $\mathcal{D}_n^{\text{opt}}$, but since here they are only discretized, it will be sometimes necessary to increase slightly the interval with a small constant $\delta > 0$ to get a guarantee in probability

Let us now discuss some interesting choices for the first two points.

The easiest approach to apprehend is discussed in Fan et al. (2023), where they first propose to consider a loss function consisting of the average width of the prediction interval, which is an increasing function of the simple loss $L\left(f, (\boldsymbol{X}_i, Y_i)_{i \in \mathcal{D}_n^{\text{opt}}}\right) = \frac{1}{n_{\text{opt}}} \sum_{i \in \mathcal{D}_n^{\text{opt}}} f(\boldsymbol{X}_i)$. Intuitively

this makes sense, since we would like to have the narrower intervals such that the coverage is guaranteed. They also consider a specific case where $\mathcal{F}$ is finite, and consists of non-negative linear combinations of pretrained estimators $\hat{f}^j_{\mathcal{D}_n^{\text{ptrain}}}$, $j = 1, \ldots, J$ on the proper training set $\mathcal{D}_n^{\text{ptrain}}$, such as quantile regression of squared residuals at different levels with different methods (splines, random forests, xgboost, neural networks, ...). The optimization then writes

$$\begin{aligned} \min_{\gamma_1,\ldots,\gamma_J \geq 0} \quad & \frac{1}{\mathcal{D}_n^{\text{opt}}} \sum_{i \in \mathcal{D}_n^{\text{opt}}} \sum_{j=1}^{J} \gamma_j \hat{f}^j_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i) \\ \text{s.t.} \quad & \sum_{j=1}^{J} \gamma_j \hat{f}^j_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i) \geq \left(Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)\right)^2, \ \forall i \in \mathcal{D}_n^{\text{opt}} \end{aligned}$$

which is a constrained linear programming problem which can be solved efficiently, the non-negativity constraints guaranteeing that the optimal function will be non-negative (recall that it is supposed to estimate the conditional variance).

Another prominent class of functions $\mathcal{F}$, discussed in both Liang (2022) and Fan et al. (2023), consists of functions in a reproducing kernel Hilbert space (RKHS), but with a subtlety since f must be non-negative. Interestingly, groundbreaking recent work on kernel-based models for non-negative functions was developed by Marteau-Ferey et al. (2020), inspired by older results on so-called *sum-of-squares*. More precisely, suppose $\mathcal{F}$ is an RKHS of functions with kernel k, feature map $\phi(\boldsymbol{x}) = k(\boldsymbol{x}, \cdot)$ and scalar product $\langle \cdot, \cdot \rangle$. Marteau-Ferey et al. (2020) propose to consider functions f written as the quadratic form

$$f(\boldsymbol{x}) = \langle \phi(\boldsymbol{x}), A[\phi](\boldsymbol{x}) \rangle$$

for a bounded Hermitian linear operator $A : \mathcal{F} \to \mathcal{F}$ and, if we assume it is semi-definite positive, the resulting function f is non-negative. This setting is particularly useful for the following facts (among others):

- The regularity of f can be controlled by standard operator norms on A: for example the squared Frobenius norm $\|A\|_F^2$ is the equivalent of the ridge penalty, and the nuclear norm $\|A\|_\star$ favors low-rank solutions similarly to the lasso
- When writing an optimization problem over this space of functions, if the objective function consists of the sum of a continuous loss function and a regularizer built with the two aforementioned norms, the solution admits an explicit and finite representation via a representer theorem

This powerful representer theorem states that the optimal operator A_* can be expressed as $A_* = \sum_{i=1}^n \sum_{j=1}^n B_{ij} \phi(\boldsymbol{X}_i) \otimes \phi(\boldsymbol{X}_j)$ for some semi-definite positive matrix $B \succeq 0$ and data points $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, yielding an optimal function f_* given by $f_*(\boldsymbol{x}) = \sum_{i=1}^n \sum_{j=1}^n B_{ij} k(\boldsymbol{x}, \boldsymbol{X}_i) k(\boldsymbol{x}, \boldsymbol{X}_j)$. Equipped with this results, we can efficiently re-write all our quantities of interest:

- $f(\boldsymbol{X}_i)$ appearing in the constraints is equal to $f(\boldsymbol{X}_i) = \langle K_i, BK_i \rangle$

- $\sum_i f(\boldsymbol{X}_i) = \text{tr}(KBK)$
- $\|A\|_F^2 = \text{tr}(KBKB)$
- $\|A\|_\star = \text{tr}(KB)$

where K is the gram matrix with entries $K_{ij} = k(\boldsymbol{X}_i, \boldsymbol{X}_j)$. This means that our initial optimization problem (2.2) restricted to this class of functions may be written as

$$\begin{aligned} \min_{B \succeq 0} \quad & \nu_1 \text{tr}(KBK) + \nu_2 \text{tr}(KBKB) + \nu_3 \text{tr}(KB) \\ \text{s.t.} \quad & \langle K_i, BK_i \rangle \geq \left(Y_i - \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{X}_i)\right)^2, \ \forall i \in \mathcal{D}_n^{\text{opt}} \end{aligned} \tag{2.3}$$

if we restrict ourselves to an objective function mixing interval width and regularity for some constants $\nu_1, \ \nu_2, \ \nu_3 \geq 0$. This is a semi-definite program for which we have efficient solvers, but a specific one for large n was also proposed in Marteau-Ferey et al. (2020). Liang (2022) considers $\nu_1 = \nu_2 = 0$ and $\nu_3 = 1$, while Fan et al. (2023) considers $\nu_1 = 1$, $\nu_2 = 0$ and the nuclear norm with a threshold constraint r, which is equivalent to some $\nu_3 > 0$. We can now state a coverage guarantee for such a procedure.

Theorem 12 (100% conditional coverage guarantee with kernels - Fan et al. (2023)). Suppose that $\sup_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}) \leq b$, then for any $\delta > 0$ the prediction interval

$$\hat{C}^{100\%}_{\mathcal{D}_n^{\text{pt-opt}}}(\boldsymbol{x}) = \hat{\mu}_{\mathcal{D}_n^{\text{ptrain}}}(\boldsymbol{x}) \pm \sqrt{\hat{f}_{\mathcal{D}_n^{\text{opt}}}(\boldsymbol{x}) + \delta}$$

where $\hat{f}_{\mathcal{D}_n^{\text{opt}}}$ is the solution of (2.3) with $\nu_1 = 1$, $\nu_2 = \nu_3 = 0$ and a constraint $\text{tr}(KB) \leq r$ satisfies

$$\mathbb{P}\left\{\mathbb{P}\left(Y_{n+1} \in \hat{C}^{100\%}_{\mathcal{D}_n^{\text{pt-opt}}}(\boldsymbol{X}_{n+1}) | \mathcal{D}_n^{\text{pt-opt}}\right) \geq 1 - \frac{32r\sqrt{b}}{\delta\sqrt{n}}\sqrt{\mathbb{E}[k(\boldsymbol{X}, \boldsymbol{X})]} - \sqrt{\frac{2t}{n}}\right\} \geq 1 - e^{-t}$$

where $\mathcal{D}_n^{\text{pt-opt}} = \mathcal{D}_n^{\text{ptrain}} \cup \mathcal{D}_n^{\text{opt}}$.

Remark. To elaborate further on this result:

- This is a distribution-free and finite-sample conditional statement, but with some constants: in previous theorems you may recall constants related to algorithm stability, here they are related to the complexity of the class of functions considered. For example, notice that the lower the rank r, the closer we get to 1. Also note that for typical stationary kernels we have $b = \mathbb{E}[k(\boldsymbol{X}, \boldsymbol{X})] = 1$
- With additional weak distributional assumptions, Liang (2022) obtains very similar guarantees

- As a side note, they do not use exchangeability in the proof: their results may be more easily generalizable to other settings than conformal prediction

For the full procedure, we still need to discuss two points:

- The final choice of $\hat{\lambda}^{\alpha}_{\mathcal{D}_n^{\text{cal}}}$ is made on an independent dataset $\mathcal{D}_n^{\text{cal}}$ to achieve the target coverage $1-\alpha$ by computing a quantile, very similarly to what is done with split conformal. Training conditional validity of the final intervals is proved in Liang (2022) and Fan et al. (2023), but at the cost of additional weak distributional assumptions (e.g. control of the tails of the conditional distribution, or absolute continuity). We do not report the results here since the notations may be heavy, but let us mention one more time that they are not based on exchangeability

- I decided here to present the approach which considers three independent datasets to build the final interval, but actually Liang (2022); Fan et al. (2023) prove results when both $\hat{\mu}$ and $\hat{f}$ are trained jointly in a unique optimization problem, such that in the end we only need $\mathcal{D}_n^{\text{ptrain}}$ and $\mathcal{D}_n^{\text{cal}}$, just like for split conformal

To conclude on these methods, they are still novel and have not been as thoroughly investigated as conformal prediction, but they appear promising in the way they recast the goal as a training optimization procedure.

2.3.2 Test conditional coverage

Coming back to our initial discussion on local adaptivity, it is finally the time to define mathematically what it precisely entails. It should be intuitive now that the cause for non-adaptive intervals lies in the fact that their coverage is guaranteed when averaged over all test points, while we would like guarantees that are valid when the test point changes. This means that we actually seek prediction intervals $\hat{C}_{\mathcal{D}_n}$ verifying the following conditional guarantee:

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \boldsymbol{X}_{n+1} = \boldsymbol{x}\right) \geq 1 - \alpha \tag{2.4}$$

for almost all $\boldsymbol{x} \in \mathcal{X}$ under $P_{\boldsymbol{X}}$, where the probability is over $\mathcal{D}_n$ and Y_{n+1}. This is called **test conditional coverage**, or X-conditional coverage, or object conditional coverage.

In the case of distribution-free inference, we want (2.4) to hold for all possible distributions $P_{\boldsymbol{X}Y}$. Unfortunately conformal prediction cannot achieve this, as well as any other method, due to a famous negative theorem.

Theorem 13 (Impossibility of test conditional coverage - Vovk (2012); Barber et al. (2021*a*)). Suppose that $\hat{C}_{\mathcal{D}_n}$ satisfies (2.4) for almost all $\boldsymbol{x} \in \mathcal{X}$ under $P_{\boldsymbol{X}}$ and all distributions $P_{\boldsymbol{X}Y}$. Then, for all distributions $P_{\boldsymbol{X}Y}$,

$$\mathbb{E}\left[\lambda\left(\hat{C}_{\mathcal{D}_n}(\boldsymbol{x})\right)\right] = \infty$$

at almost all points $\boldsymbol{x}$ aside from the atoms of $P_{\boldsymbol{X}}$ and where λ is the Lebesgue measure.

Remark. Looking closely at this theorem, remark that:

1. This means that the prediction interval has infinite expected length, hence the so-called impossibility
2. For the case where $P_{\boldsymbol{X}}$ is purely atomic, i.e. if $\mathcal{X}$ is discrete, we can get test conditional coverage very easily since we avoid the negative result. This is as simple as running conformal prediction separately for each discrete value $\boldsymbol{x}$

Starting from this bad news, it is obvious that some of the requirements must be lessened in order to get test conditional coverage in practice (but in an approximate sense). There are several recent point of views, that we only discuss briefly below.

Relaxing the conditioning. Instead of asking (2.4) to hold for almost all $\boldsymbol{x} \in \mathcal{X}$ under $P_{\boldsymbol{X}}$, the first attempt towards approximate test conditional coverage simply and naturally relies on binning. Consider a fixed partition $\mathcal{X} = \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_K$ of $\mathcal{X}$ into K disjoint sets, then by running standard conformal prediction on each $\mathcal{X}_k$ separately gives the guarantee

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \boldsymbol{X}_{n+1} \in \mathcal{X}_k\right) \geq 1 - \alpha$$

for all $k = 1, \ldots, K$, see Vovk (2012). A more recent proposal by Barber et al. (2021*a*) focuses on conditioning locally around a point $\boldsymbol{x}$, i.e. they propose a guarantee given by

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \boldsymbol{X}_{n+1} \in B(\boldsymbol{x}, r)\right) \geq 1 - \alpha$$

for all $\boldsymbol{x} \in \mathcal{X}$ such that $\mathbb{P}(\boldsymbol{X} \in B(\boldsymbol{x}, r)) \geq \delta$ where $B(\boldsymbol{x}, r)$ is the ℓ_2 ball of radius r centered at $\boldsymbol{x}$.

The drawback of this type of approaches is that they require a large number of calibration samples to be robust (such that after conditioning there are still enough samples to compute quantiles). In addition, binning is strongly impacted by the curse of dimensionality.

Relaxing the distribution-free requirement. Another obvious path for approximate test conditional coverage consists in allowing (2.4) to depend on $P_{\boldsymbol{X}Y}$ instead of being valid for any distribution. A very recent result by Deutschmann et al. (2024) show that if $\hat{C}_{\mathcal{D}_n}(\boldsymbol{x}) =$

$\left\{y \in \mathbb{R} :\ S(\boldsymbol{X}, y) \leq \hat{q}^{+}_{n,\alpha}\left\{S_i,\ i \in \mathcal{D}^{\text{cal}}_n\right\}\right\}$ is built from a score function S with split conformal, then

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1}) | \boldsymbol{X}_{n+1} \in B\right) \geq 1 - \alpha - \frac{\sqrt{1 - \exp(-I(\boldsymbol{X}, S(\boldsymbol{X}, Y)))}}{\delta}$$

for any set $B \in \mathcal{X}$ such that $\mathbb{P}(\boldsymbol{X} \in B) \geq \delta$ and where $I(\cdot, \cdot)$ denotes the mutual information. Observe first that, as expected, the smaller δ, the harder it is to get the target coverage (see also Barber et al. (2021*a*)). But interestingly, it is possible to counterbalance this effect by reducing as much as possible the numerator of the last term. Indeed, this numerator has a lower bound equal to 0, which is attained when the mutual information is 0, i.e. when $\boldsymbol{X}$ and $S(\boldsymbol{X}, Y)$ are independent. This paves the way for learning a score function specifically designed for the distribution at hand, similarly to what we discussed in the previous section, but with a criterion based on mutual information.

Let us also mention another recent work by Plassier et al. (2024), where they propose a prediction interval based on an estimate of the conditional distribution which leads to test conditional coverage with a penalty involving the total variation distance between this estimate and the true conditional distribution. Although their result is based on different ideas, there may be subtle links given that mutual information and total variation distance are similar.

Relaxing the probability. Finally, a completely original line of work was initiated by Gibbs et al. (2023), where the authors remark that requiring (2.4) with an equality for almost all $\boldsymbol{x} \in \mathcal{X}$ under $P_{\boldsymbol{X}}$ is theoretically equivalent to

$$\mathbb{E}\left[f(\boldsymbol{X}_{n+1})\left(\mathbb{1}_{Y_{n+1} \in \hat{C}_{\mathcal{D}_n}(\boldsymbol{X}_{n+1})} - (1-\alpha)\right)\right] = 0$$

for all measurable functions f. Their idea is then to relax the last part, by requiring this guarantee to hold only for all functions $f \in \mathcal{F}$ for a specific function class $\mathcal{F}$. The simple choice of constant functions leads to marginal coverage, while the set of all measurable functions leads to the test conditional one, but is unachievable. Other choices of function classes with intermediate complexity between these two lead to approximate test conditional coverage, with an explicit construction of the corresponding prediction intervals. Gibbs et al. (2023) discuss for example linear functions, but also Lipschitz functions and functions in a RKHS. Striking is the fact that choosing $\mathcal{F}$ amounts to solving a supervised learning optimization problem over $\mathcal{F}$, in the same vein as what we discussed previously for Liang (2022); Fan et al. (2023): there may be links between these two ideas although they originated from highly different starting points.

2.4 Concluding remarks

During this course, we have introduced conformal prediction and related methods to provide prediction intervals with the goal of assessing uncertainty in machine learning predictions with theoretical guarantees on their coverage (see a summary in the next page). This is a highly active research area, with many new contributions each year, and we just mentioned a very

small portion of the abundant bibliography (and keep in mind that we did not even address classification problems).

Before ending this lecture, I would like to mention a few typical specificities of computer experiments, that should make you think a little before diving blindly in conformal prediction:

- The absence of noise on the target: you are all aware of this fact, but data collected on physical simulators are not noisy (except for stochastic simulator of course), so you should take a step back to interpret the coverage guarantees. In such a case for example, if you want to aim for both training and test conditional coverage, what randomness remains? Even without going as far as that, recall that we have seen several times an absolute continuity assumption on the target conditional probability, which is obviously violated in such a case
- The data very often come from an optimized design of experiments, with a spatial structure which violates both i.i.d. and exchangeability assumptions. So in practice, even if from a numerical point of view you may observe your target coverage, at this time it is impossible to get any theoretical guarantee

But do not despair: this only means there is a lot of stimulating research work to do in order to be able to use the full potential of such techniques in computer experiments!

Method	Marginal coverage	Training cond. coverage	Centered	Requires re-training	Varying width	Comments
Full conformal	$1-\alpha$	$\to 1-\alpha$ $(\mathcal{SA}_1^{\text{in}})$	no	yes	yes	Very high computational cost
Split conformal	$1-\alpha$	$1-\alpha$	yes	no	no	Loss of efficiency due to splitting
Jackknife	$\approx 1-\alpha$ $(\mathcal{AC}^{\text{e}})$, $(\mathcal{SA}_1)$	$\to 1-\alpha$ $(\mathcal{AC}^{\text{b}})$, $(\mathcal{SA}_2)$	yes	yes	no	High computational cost
Jackknife+	$1-2\alpha$ $\approx 1-\alpha$ $(\mathcal{AC}^{\text{e}})$, $(\mathcal{SA}_1)$	$\to 1-\alpha$ $(\mathcal{AC}^{\text{e}})$, $(\mathcal{SA}_2)$	no	yes	$\approx$ no	High computational cost
CV+	$\approx 1-2\alpha$	$\approx 1-2\alpha$ large m, $n = Km$	no	yes	$\approx$ no	Moderate computational cost
CQR	$1-\alpha$	$1-\alpha$	no	no	yes	Loss of efficiency due to splitting Requires QR software
RKHS	$\to 1-\alpha$ $(\mathcal{AC})$-like	$\to 1-\alpha$ $(\mathcal{AC})$-like	yes	no	yes	Loss of efficiency due to splitting Requires SDP software

Bibliography

Amann, N., Leeb, H. and Steinberger, L. (2023), 'Assumption-lean conditional predictive inference via the jackknife and the jackknife+', *arXiv preprint arXiv:2312.14596* .

Angelopoulos, A. N. and Bates, S. (2021), 'A gentle introduction to conformal prediction and distribution-free uncertainty quantification', *arXiv preprint arXiv:2107.07511* .

Barber, R. F. (2024), 'Conformal prediction tutorial'.
URL: *https://rinafb.github.io/talks/*

Barber, R. F., Candes, E. J., Ramdas, A. and Tibshirani, R. J. (2021*a*), 'The limits of distribution-free conditional predictive inference', *Information and Inference: A Journal of the IMA* **10**(2), 455–482.

Barber, R. F., Candès, E. J., Ramdas, A. and Tibshirani, R. J. (2021*b*), 'Predictive inference with the jackknife+', *The Annals of Statistics* **49**(1), 486 – 507.

Barber, R. F., Candes, E. J., Ramdas, A. and Tibshirani, R. J. (2023), 'Conformal prediction beyond exchangeability', *The Annals of Statistics* **51**(2), 816–845.

Bian, M. and Barber, R. F. (2023), 'Training-conditional coverage for distribution-free predictive inference', *Electronic Journal of Statistics* **17**(2), 2044–2066.

Bousquet, O. and Elisseeff, A. (2002), 'Stability and generalization', *The Journal of Machine Learning Research* **2**, 499–526.

Chen, W., Wang, Z., Ha, W. and Barber, R. F. (2016), 'Trimmed conformal prediction for high-dimensional models', *arXiv preprint arXiv:1611.09933* .

Chernozhukov, V., Wüthrich, K. and Zhu, Y. (2021), 'Distributional conformal prediction', *Proceedings of the National Academy of Sciences* **118**(48), e2107794118.

Deshpande, S., Marx, C. and Kuleshov, V. (2024), Online calibrated and conformal prediction improves bayesian optimization, *in* 'International Conference on Artificial Intelligence and Statistics', PMLR, pp. 1450–1458.

Deutschmann, N., Rigotti, M. and Martinez, M. R. (2024), 'Adaptive conformal regression with split-jackknife+ scores', *Transactions on Machine Learning Research* .

Fan, J., Ge, J. and Mukherjee, D. (2023), 'Utopia: Universally trainable optimal prediction intervals aggregation', *arXiv preprint arXiv:2306.16549* .

Fannjiang, C., Bates, S., Angelopoulos, A. N., Listgarten, J. and Jordan, M. I. (2022), 'Conformal prediction for the design problem', *arXiv preprint arXiv:2202.03613* .

Gibbs, I., Cherian, J. J. and Candès, E. J. (2023), 'Conformal prediction with conditional guarantees', *arXiv preprint arXiv:2305.12616* .

Izbicki, R., Shimizu, G. T. and Stern, R. B. (2019), 'Flexible distribution-free conditional predictive bands using density estimators', *arXiv preprint arXiv:1910.05575* .

Jaber, E., Blot, V., Brunel, N., Chabridon, V., Remy, E., Iooss, B., Lucor, D., Mougeot, M. and Leite, A. (2024), 'Conformal approach to gaussian process surrogate evaluation with coverage guarantees', *arXiv preprint arXiv:2401.07733* .

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J. and Wasserman, L. (2018), 'Distribution-free predictive inference for regression', *Journal of the American Statistical Association* **113**(523), 1094–1111.

Liang, R. and Barber, R. F. (2023), 'Algorithmic stability implies training-conditional coverage for distribution-free prediction methods', *arXiv preprint arXiv:2311.04295* .

Liang, T. (2022), 'Universal prediction band via semi-definite programming', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **84**(4), 1558–1580.

Marteau-Ferey, U., Bach, F. and Rudi, A. (2020), 'Non-parametric models for non-negative functions', *Advances in neural information processing systems* **33**, 12816–12826.

Plassier, V., Fishkov, A., Panov, M. and Moulines, E. (2024), 'Conditionally valid probabilistic conformal prediction', *arXiv preprint arXiv:2407.01794* .

Romano, Y., Patterson, E. and Candes, E. (2019), 'Conformalized quantile regression', *Advances in neural information processing systems* **32**.

Soloff, J. A., Barber, R. F. and Willett, R. (2024), 'Bagging provides assumption-free stability', *Journal of Machine Learning Research* **25**(131), 1–35.

Stanton, S., Maddox, W. and Wilson, A. G. (2023), Bayesian optimization with conformal prediction sets, *in* 'International Conference on Artificial Intelligence and Statistics', PMLR, pp. 959–986.

Steinberger, L. and Leeb, H. (2023), 'Conditional predictive inference for stable algorithms', *The Annals of Statistics* **51**(1), 290–311.

Tibshirani, R. J. (2024), 'Advanced topics in statistical learning: Spring 2024'.
URL: *https://www.stat.berkeley.edu/ ryantibs/statlearn-s24/lectures/conformal.pdf*

Tibshirani, R. J., Foygel Barber, R., Candes, E. and Ramdas, A. (2019), 'Conformal prediction under covariate shift', *Advances in neural information processing systems* **32**.

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

Vovk, V., Gammerman, A. and Shafer, G. (2005), *Algorithmic learning in a random world*, Vol. 29, Springer.

Vovk, V., Nouretdinov, I., Manokhin, V. and Gammerman, A. (2018), Cross-conformal predictive distributions, *in* 'conformal and probabilistic prediction and applications', PMLR, pp. 37–51.

Wang, Y., Wu, H. and Nettleton, D. (2023), 'Stability of random forests and coverage of random-forest prediction intervals', *Advances in Neural Information Processing Systems* **36**.

Xu, H., Caramanis, C. and Mannor, S. (2011), 'Sparse algorithms are not stable: A no-free-lunch theorem', *IEEE transactions on pattern analysis and machine intelligence* **34**(1), 187–193.

Zaffran, M., Féron, O., Goude, Y., Josse, J. and Dieuleveut, A. (2022), Adaptive conformal predictions for time series, *in* 'International Conference on Machine Learning', PMLR, pp. 25834–25866.