Adaptive Conformal Predictions for Time Series

Richard Boggio
Samy Mdihi
Marc Veron-Tarabeux

Table of Contents

Introduction

- Uncertainty quantification of predictive models is crucial in decision-making problems.
- Conformal prediction estimates a prediction interval in regression problems and a set of classes in classification problems.
- Aggregation of experts
- Generation of intervals along different horizons

Table of Contents

Methods

- Gaussian

$$C_\alpha := [\hat{y} \pm Q_{1-\alpha} * Std]$$

- CP

$$C_\alpha := [\hat{y} \pm Q_{1-\alpha}(|\hat{y} - y|)]$$

- ACP

$$\begin{cases} C_{\alpha t} := [\hat{y} \pm Q_{1-\alpha t}(|\hat{y} - y|)] \\ \alpha_{t+1} = \alpha_t + \gamma(\alpha - \mathbb{1} y_t \notin C_{\alpha t}) \end{cases}$$

Table of Contents

Interval Class

- Class Interval in conf
  - This is the mother class for all intervals
  - Users are not expected to use this class as-is. Instead, user should use it in other modules like time series for example, but can be extended to other regression type problems
- We added an __all__list in the init files so that the classes added can be used everywhere

Conf

We added some methods to be able to generate some intervals

- Class Gaussian
  - Function update
    If the number of residuals is equal to the window size we calculate the interval
    half_inter $= norm.ppf(self.alpha) * \sqrt{self.var.get()}$
  - Function get
    To return the current value of the interval
- AC and ACP...

Time Series

The script evaluates.py has been modified and we added intervals.py

- intervals.py
  - Class HorizonInterval
    It is the mother class for the below one
  - ForecastingInterval
    It updates the prediction interval along the horizon : measures the prediction interval at each time step ahead
- evaluates.py
  - The parameter interval has been added in every function. For example iter_evaluate : Get first residuals series with the pre-trained model

Table of Contents