POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A Complete Suite for Conformal Prediction of Simple and Complex Data in R, with some theoretical extensions

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Paolo Vergottini

Advisor: Prof. Simone Vantini

Co-advisors: Dott. Jacopo Diquigiovanni Dott. Matteo Fontana Prof. Aldo Solari

Academic year: 2020-2021

1. Introduction

Conformal Prediction methods allow to construct distribution-free prediction sets in a regression framework, requiring only *i.i.d.* regression data. They are extremely flexible, not relying on specific regression models, as well as effective, defining prediction sets with at least marginal coverage level $1-\alpha$. A broad review of the theory of conformal prediction can be found in [5]. Python is the most common framework for implementing these methods. For example, the **nonconformist** and **libconform** libraries handle both classification and prediction for regression with univariate responses. For R users there is only one solid package available: **conformalInference**[1], which deals with the univariate case.
The aim of our work is twofold: first, we want to provide high-quality official code for conformal on R, and secondly, we want to develop methods, missing also in Python, for more complex regression frameworks. Therefore, we implemented conformal prediction for multivariate response (with **conformalInference.multi**) as well as multivariate functional response (with **conformalInference.fd**[2]). We also incorporated novel extensions of Jackknife+ and Multi Split conformal methods, originally designed for univariate frameworks, along with Full Conformal and Split Conformal prediction methods. Additionally, we also proposed prof. implementations of conformal prediction methods to enrich the package **conformalInference**.

2. Conformal prediction

In the following discussion we will consider a set of values $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \sim i.i.d.$ [3], and define $z_i = (x_i, y_i)\ \forall i$. We will discuss three responses:

1. Univariate response: $y \in R$
2. Multivariate response: $y \in R^q$
3. Multivariate functional response: $y \in \prod_{j=1}^{q} L^{\infty}(\tau_j)$, where τ_j is a closed and bounded subset of $R^{d_j}, d_j \in N_{>0}$. In particular y_i is a function $i = 1, ..., q$.

A crucial element in conformal prediction theory is **non-conformity**: given a new value z_{n+1}, we

[1] The package is not yet available on CRAN and is currently hosted on GitHub. The author is Prof. Ryan Tibshirani of the Statistics and Machine Learning Department of Carnegie Mellon University.

[2] Both **conformalInference.multi** and **conformalInference.fd** are available on CRAN.

[3] The milder assumption of exchangeability could suffice.

can score how unusual it is from all the other observations $\{z_1, ..., z_n\}$ with the medium of a non-conformity measure, a real-valued function $\mathcal{A}$, returning $\mathcal{A}(\{z_1, ..., z_n\}, z_{n+1})$ and assigning greater value to the most unusual points. We considered non-conformity measures specific for each regression framework, and we allowed for local scaling of the non-conformity scores using **modulation functions**.

2.1. Full conformal

The first prediction method we will discuss is **Full Conformal**. Essentially, for a given test observation x_{n+1}, it ranks how well a candidate point $z_{n+1} = (x_{n+1}, y)$ matches with all the rest of the data, selecting y from a grid of candidates. This approach can be extended to the multivariate response case, as long as q is small, avoiding computational issue. However, a potential extension to the functional case suffers from a critical problem: one should consider as candidates all possible functions in $\prod_{j=1}^{q} L^{\infty}(\tau_j)$, where τ_j is a closed and bounded subset of $R^{d_j}, d_j \in N_{>0}$. Using the non-conformity measure we score the residuals:

$$R_i := \mathcal{A}(\{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n, , z_{n+1}\}, z_i)$$
$$R_{n+1} := \mathcal{A}(\{z_1, ..., z_n\}, z_{n+1})$$

Our next step is to rank R_{n+1} with respect to all the other residual scores $R_1, ..., R_n$ and calculate the fraction of values with higher non-conformity scores than the candidate point as

$$\delta_y = \frac{|j \in \{1, ..., n, n+1\} \ : R_j \geq R_{n+1}|}{n+1}$$

Lastly, we identify the full conformal prediction set as $C_{full}(x_{n+1}) = \{y \in \mathcal{Y} \ : \delta_y > \alpha\}$. Moreover, the method guarantees a coverage level $1 - 2\alpha$.

2.2. Split conformal

Another method called **Split Conformal** was suggested to overcome the high computational demand of full conformal. This method works seamlessly also in multivariate and functional response frameworks. Essentially the observations $y_1, ..., y_n$ are randomly split into a training set $\mathcal{I}_1$ and a validation set $\mathcal{I}_2$, respectively of cardinality m and l. The first set is used to train, just once, the regression model, while the second set is used to compute, for every $i \in \mathcal{I}_2$, the distance of each y_i from the fitted value $\hat{\mu}_{\mathcal{I}_1}(x_i)$ through a non-conformity measure. Indeed, consider

$$R_i := \mathcal{A}(\{z_k : k \in \mathcal{I}_1\}, z_i) \quad i \in \mathcal{I}_2$$
$$R_{n+1} := \mathcal{A}(\{z_k : k \in \mathcal{I}_1\}, z_{n+1})$$

and use the new definitions of residuals' scores to obtain δ_i as with full conformal.
Hence, $C_{split}(x_{n+1}) = \{y \in \mathcal{Y} \ : \delta_y > \alpha\}$. Once again, split conformal prediction sets are valid, but there is no guarantee of exactness, i.e. having precise coverage of $1 - \alpha$.
Split conformal, however, has two main drawbacks. First of all, the observations in the validation set are not used to train the model (in some ways we are "wasting" precious information). Secondly, due to the stochastic nature of splitting data between $\mathcal{I}_1$ and $\mathcal{I}_2$, different runs of the algorithm will provide slightly different prediction sets.

2.3. Jackknife+

The **Jackknife+** method was introduced to solve the main issues of full conformal and split conformal. It is less computational demanding than full conformal, whereas, contrary to split conformal, it exploits all the observations and returns always the same prediction sets. Furthermore, jackknife+ relies on the assumption of *i.i.d.* data and is built for the univariate response case. The jackknife+ prediction interval:

$$C_{jack+} = [q_\alpha\{\hat{\mu}_{-i}(x_{n+1}) - R_i^{LOO}\}, \\ q_{1-\alpha}\{\hat{\mu}_{-i}(x_{n+1}) + R_i^{LOO}\}]$$

where $R_i = |y_i - \hat{\mu}_{-i}(x_i)| \ \forall i = 1, ..., n$ are the absolute leave-one-out (or LOO) residuals and $\hat{\mu}$ is a regression model trained over the whole dataset $\{z_1, ..., z_n\}$. The LOO residuals can be computed by fitting a complete regression model and n leave-one-out models, removing one observation at a time for training. In addition, jackknife+ provides a guarantee of $(1 - 2\alpha)\%$ coverage for its intervals.
To extend the model to multivariate or functional cases, we exploit the concept of **non-conformity measure** $\mathcal{A}$. By ordering points according to their non-conformity score we can define a multivariate and functional quantile $(q_\alpha^{\mathcal{A}})$ as the level set induced by the non-conformity measure. As a result, in the mul-

tivariate context we could simply extend jackknife+ as :

$$C_{jack+} = \{y \in R^q : y \in [q_\alpha^{\mathcal{A}}(\{\hat{\mu}_{-i}(x_{n+1}) \pm R_i^{LOO} : i = 1, ..., n\})]\}$$

While in the functional context:

$$C_{jack+} = \{y \in \prod_{j=1}^{q} L^\infty(\tau_j) : y(t) \in [q_\alpha^{\mathcal{A}}(\{\hat{\mu}_{-i}(x_{n+1}) \pm R_i^{LOO} : i = 1, ..., n\})(t)] \; \forall t \in \prod_{j=1}^{q} \tau_j\}$$

After obtaining the level sets with $q_\alpha^{\mathcal{A}}$, we compute the axis-aligned minimum bounding box (or AABB[4]), effectively projecting the prediction region over the axis. The method yields a prediction interval for each component of the response, simplifying the interpretation of the results.

2.4. Multi split conformal

Among the drawbacks of split conformal is the inherent randomness in the splitting procedure, since each split produces a valid prediction set. To overcome this limitation, the **Multi Split Conformal** method was developed. Here, the split conformal method is run multiple times (B) and then the prediction intervals are combined. A complete formulation of this method can be found in [3]. To determine the minimum number of intervals within which a point must be contained to be included in our final prediction set, we use τ. Formally, given the simulated split conformal intervals $C^{[1]}, ..., C^{[B]}$, we can define $\Pi^y = \frac{1}{B}\sum_{b=1}^{B} 1\{y \in C^{[b]}\} \; \forall y \in R$. Then the multi split conformal prediction interval is:

$$C_{msplit}(x_0) = \{y \in R \; : \; \Pi^y > \tau\}$$

It is important to note that the split conformal intervals $C^{[1]}, ..., C^{[B]}$ are obtained by setting the miscoverage interval to $\alpha(1 - \tau + \lambda/B)$, where λ is a smoothing parameter (a positive integer). The multi split prediction interval has coverage at least $1 - \alpha$.
The extension to the multivariate and functional case recalls the discussion for jackknife+. Indeed, one runs the split conformal methods multiple times, joins their lower bounds and upper bounds into a single set and rank data points according to a specific non-conformity measure. Then, we will set the level of the quantile as $2\tau B$, i.e. twice the chosen amount of areas or bands a point must be contained in to be part of the final prediction set. Finally we output the axis-aligned bounding box of the previous level set.

[4]More detail at `https://worddisk.com/wiki/Bounding_box/#1`

2.5. Conformalized quantile regression (CQR)

Conformalized quantile regression is an innovative prediction method, introduced in [2], which combines the theory of quantile regression and of conformal prediction. Quantile regression estimates the quantile function $q_\alpha(x) = inf\{y \in \mathbb{R} : F(y|X = x) \geq \alpha\}$, with $\hat{q}_\alpha(x)$. Intuitively we may construct a naive prediction interval of level $1-\alpha$ with the form $C_{naive}(x) := [\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}]$. In practice, thought, this choice is prone to undercoverage. However, $C_{naive}(x)$ can be exploited to define a fitting non-conformity measure

$$R_i := max(\hat{q}_{\alpha_{lo}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{up}}(X_i))$$

From this equation two variants of the *classical* non-conformity measure emerged: the *median* one, which computes $\hat{q}_{0.5}(x)$ to resize the interval, and the *scaled* version, which divides R_i for the length of the naive interval.
Then the full conformal version of CQR descends naturally, just by using one of these non-conformity measure, while the split conformal one requires to adapt the prediction intervals to take into account the various types of local scaling.

3. conformalInference

As detailed in the thesis' appendix, we contributed to the package **conformalInference** by adding three main prediction function.
The first is an implementation of jackknife+. We modified the already existing `conformal.pred.jack` function, which originally handled only the classic jackknife prediction method. Indeed, when the user sets the parameter `plus = TRUE` it performs jackknife+, otherwise it proceeds with jackknife. To speed

up computations we relied on the package **future.apply** to parallelise the code. However, whenever this package is not installed, then the code proceeds in a sequential fashion, without interrupting the process abruptly.
Secondly, we implemented multi split conformal prediction, which runs the split conformal function many times. Hence, we employed once again parallelisation with **future.apply**.
Finally we included two versions of CQR: a full conformal one and a split conformal one, as described in subsection 2.5.

4. conformalInference.multi

The **conformalInference.multi** package performs conformal prediction for regression when the response variable y is multivariate. The modular structure of the package allows the user to employ custom-coded regression functions, according to the specific regression task at hand.
Three different regression methods are available: the *mean model*, where the response is modeled using the sample mean of the observations, the *linear model* and the *elastic net model*, which applies the **glmnet** package and also contains lasso and ridge regression subcases.
We implemented four prediction methods: full conformal, split conformal, jackknife+ and multi split conformal. Lastly, we included a custom plot functions to display the produced prediction regions.

5. conformalInference.fd

conformalInference.fd provides conformal prediction when the response variable y is a multivariate functional datum. It shares the structure with its multivariate counterpart.
We just implemented two elementary models for functional regression: the *mean model*, where y is modeled as the mean of the of the observed responses, and a *concurrent regression model*, with the form:

$$y_k(s) = \beta_0(s) + \sum_{i=1}^{p} \beta_i(s) x_i(s) + \epsilon(s) \quad k = 1, ..., q$$

For this latter model, the evaluation grid for function $x \in \prod_{j=1}^{p} L^{\infty}(\tau_j)$ and for $y \in \prod_{j=1}^{q} L^{\infty}(\tau_j)$ must be the same.
When dealing with functional data, the input structure tends to be quite complex. To simplify the input, we developed `convert2data`, a function that converts functional data types fD, $fData$ or $mfData$ (the first type defined in package **fda** and the other two defined in package **roahd**) into lists of pointwise evaluations. Following the work detailed in section 2, we included three prediction methods: split conformal, jackknife+ and multi split conformal. Our last effort was to design a versatile plot function that takes as input `out`, the output of one of the prediction methods in the package, and displays the prediction bands.

6. Example

As a case study, we examined the BikeMi data, a mobility dataset that tracks all bike rentals for the BikeMi service active in the city of Milan. This was an opportunity to present a practical application of our packages. Therefore we transformed the raw dataset to perform an analysis on multivariate response regression and on multivariate functional response regression. We based the models on the study contained in [4]. After describing the actual code, we compared the results of the prediction methods on the basis of three factors: the size of the prediction regions, the computation time, and the empirical coverage.

7. Conclusions

The article recaps the main concepts of Conformal Prediction theory and proposes extensions to the multivariate and functional frameworks of multi split conformal and jackknife+ prediction methods. Next, the structure and the main functions of **conformalInference.multi** and **conformalInference.fd** are extensively discussed. The last section of the paper presents a case study using mobility data collected by BikeMi. Finally, our contributions to the package **conformalInference** are presented in the appendix.
Therefore, we have bridged the gap between R and other programming languages through the introduction of conformal inference tools for regression in multivariate and functional contexts, and as a result shed light on how versatile as well as effective these methods can be.
We envision two future directions for our work. To begin with, in accordance with its author, we would like to submit **conformalInference**

for publication on CRAN, which would enrich the pool of distribution-free prediction methods available to R users. Secondarily, we would be interested in expanding on the work presented in [1], to explore conformal inference prediction tools in time series analysis.

References

[1] Jacopo Diquigiovanni, Matteo Fontana, and Simone Vantini. Distribution-free prediction bands for multivariate functional time series: an application to the italian gas market, 07 2021.

[2] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[3] Aldo Solari and Vera Djordjilović. Multi split conformal prediction. *Statistics Probability Letters*, 184:109395, 2022.

[4] Agostino Torti, Alessia Pini, and Simone Vantini. Modelling time-varying mobility flows using function-on-function regression: Analysis of a bike sharing system in the city of milan. *Journal of the Royal Statistical Society.Series C: Applied Statistics*, 70(1):226–247, 11 2020.

[5] Gianluca Zeni, Matteo Fontana, and Simone Vantini. Conformal prediction: a unified review of theory and new challenges. *ArXiv*, abs/2005.07972, 2020.