Copula-based conformal prediction for multi-target regression

Soundouss Messoudi, Sébastien Destercke, Sylvain Rousseau

Copula-based conformal prediction for Multi-Target Regression

Soundouss Messoudi[a,*], Sébastien Destercke[a], Sylvain Rousseau[a]

[a] *HEUDIASYC - UMR CNRS 7253, Université de Technologie de Compiègne*
57 avenue de Landshut, 60203 COMPIEGNE CEDEX - FRANCE

Abstract

There are relatively few works dealing with conformal prediction for multi-task learning issues, and this is particularly true for multi-target regression. This paper focuses on the problem of providing valid (i.e., frequency calibrated) multi-variate predictions. To do so, we propose to use copula functions for inductive conformal prediction, and illustrate our proposal by applying it to deep neural networks and random forests. We show that the proposed method ensures efficiency and validity for multi-target regression problems on various data sets.

Keywords: Inductive conformal prediction; Copula functions; Multi-target regression; Deep neural networks; Random forests.

1. Introduction

The most common supervised task in machine learning is to learn a single-task, single-output prediction model. However, such a setting can be ill-adapted to some problems and applications.

On the one hand, producing a single output can be undesirable when data is scarce and when producing reliable, possibly set-valued predictions is important (for instance in the medical domain where examples are very hard to collect for specific targets, and where predictions are used for critical decisions). Such an

*Corresponding author
Email address: soundouss.messoudi@hds.utc.fr (Soundouss Messoudi)
URL: https://www.hds.utc.fr/ (Soundouss Messoudi)

issue can be solved by using conformal prediction approaches [1]. It was initially proposed as a transductive online learning approach to provide set predictions (in the classification case) or interval predictions (in the case of regression) with a statistical guarantee depending on the probability of error tolerated by the user, but was then extended to handle inductive processes [2]. On the other hand, there are many situations where there are multiple, possibly correlated output variables to predict at once, and it is then natural to try to leverage such correlations to improve predictions. Such learning tasks are commonly called Multi-task in the literature [3].

Most research work on conformal prediction for multi-task learning focuses on the problem of multi-label prediction [4, 5], where each task is a binary classification one. Conformal prediction for multi-target regression has been less explored, even though it can be quite useful in practice, for instance to accurately predict the localization of an object in 2D [6] or of a drone in 3D [7]. Only a few studies deal with conformal prediction for multi-target regression : Kuleshov *et al.* [8] provide a theoretical framework to use conformal predictors within manifold (e.g., to provide a mono-dimensional embedding of the multi-variate output), while Neeven and Smirnov [9] use a straightforward multi-target extension of a conformal single-output k-nearest neighbor regressor [10] to provide weather forecasts. However, this latter essentially verifies validity (i.e., having well-calibrated outputs) for each individual target. Recently, we proposed a simple method to have an approximate validity for the multi-variate prediction [11], that generally provided overly conservative results.

In this paper, we propose a new conformal prediction method fitted to multi-target regression, that makes use of copulas [12] (a common tool to model dependence between multi-variate random variables) to provide valid multi-variate predictions. The interest of such a framework is that it remains very easy to apply while linking multi-variate conformal predictions to the theoretically sound framework that are copulas. Experiments also show that it works quite well, and allows to improve upon previous heuristics [11].

Section 2 provides a general overview of our problem: a brief introduction to

conformal prediction and multi-target regression will be presented in Sections 2.1 and 2.2, before raising the problematic of applying conformal prediction to the multi-target regression setting in Section 2.3. We will then present our setting in Section 3: we will first recall the needed basic principles and theorems of copulas in Section 3.1, before detailing our conformal multi-target approach in Section 3.2. The experiments and their results are described in Section 4.

2. Inductive conformal prediction (ICP) for Multi-Target Regression

This section recalls the basics of inductive conformal regression and multi-target regression, before introducing the issues we will tackle in this paper.

2.1. Inductive conformal regression

In regression tasks, conformal prediction is a method that provides a statistical guarantee to the predictions by giving an interval prediction instead of a point prediction in the regression case. By statistical guarantee, it is meant that the set-valued predictions cover the true value with a given frequency, i.e., they are calibrated. It was first introduced as a transductive online learning approach [13] and then adapted to the inductive framework [2] where one uses a model induced from training examples to get conformal predictions for the new instances. The two desirable features in conformal regressors are (a) *validity*, i.e. the error rate does not exceed ϵ for each chosen confidence level $1-\epsilon$, and (b) *efficiency*, meaning prediction intervals are as small as possible.

Let $\wr z_1 = (x_1, y_1), z_2 = (x_2, y_2), \ldots, z_n = (x_n, y_n)\wr$ be the successive pairs of an object $x_i \in X$ and its real-valued label $y_i \in \mathbb{R}$, which constitute the observed examples. Assuming that the underlying random variables are exchangeable (a weaker condition than i.i.d.), we can predict $y_{n+1} \in \mathbb{R}$ for any new object $x_{n+1} \in X$ by following the inductive conformal framework.

The first step consists of splitting the original data set $Z = \wr z_1, \ldots, z_n\wr$ into a *training set* $Z^{tr} = \wr z_1, \ldots, z_l\wr$ and a *calibration set* $Z^{cal} = \wr z_{l+1}, \ldots, z_n\wr$, with $|Z^{cal}| = n - l$. Then, an *underlying algorithm* is trained on Z^{tr} to obtain

the *non-conformity measure* A_l, a measure that evaluates the strangeness of an example compared to other examples of a bag, called the non-conformity score. Hence, we can calculate the non-conformity score α_k for an example z_k compared to the other examples in the bag $\wr z_1, \ldots, z_l \int$ with $\alpha_k = A_l(\wr z_1, \ldots, z_l \int, z_k)$.

By computing the non-conformity score α_i for each example z_i of Z^{cal} using this equation, we get the sequence $\alpha_{l+1}, \ldots, \alpha_n$. When making a prediction for a new example x_{n+1}, we use the underlying algorithm to associate to any possible prediction $\hat{y}$ its non-conformity score $\alpha_{n+1}^{\hat{y}}$, and calculate its *p-value* which indicates the proportion of less conforming examples than z_{n+1}, with:

$$p(\hat{y}_{n+1}) = \frac{|\{i = l+1, \ldots, n, n+1 : \alpha_i \geq \alpha_{n+1}^{\hat{y}}\}|}{n - l + 1}. \tag{1}$$

The final step before producing the conformal prediction consists of choosing the *significance level* $\epsilon \in (0, 1)$ to get a prediction set with a *confidence level* of $1 - \epsilon$, which is the statistical guarantee of coverage of the true value y_{n+1} by the interval prediction $\hat{\mathbf{y}}_{n+1}$ such that

$$\hat{\mathbf{y}}_{n+1} = \{\hat{y}_{n+1} \in \mathbb{R} : p(\hat{y}_{n+1}) > \epsilon\}.$$

The most basic non-conformity measure in a regression setting is the absolute difference between the actual value y_i and the predicted value $\hat{y}_i$ by the underlying algorithm. The non-conformity score is then calculated as follows:

$$\alpha_i = |y_i - \hat{y}_i|. \tag{2}$$

The sequence of non-conformity scores $\alpha_{l+1}, \ldots, \alpha_n$ for all examples in Z^{cal} are obtained and sorted in descending order. Then, we compute the index of the $(1 - \epsilon)$-percentile non-conformity score α_s, based on the chosen significance level ϵ, such as:

$$\mathbb{P}(|y_i - \hat{y}_i| \leq \alpha_s) \geq 1 - \epsilon. \tag{3}$$

Finally, the prediction interval for each new example x_{n+1}, which covers the true output y_{n+1} with probability $1 - \epsilon$ is calculated as:

$$\hat{\mathbf{y}}_{n+1} = [\hat{y}_{n+1} - \alpha_s, \hat{y}_{n+1} + \alpha_s]. \tag{4}$$

The drawback of this standard non-conformity measure is that all prediction intervals are equally sized ($2\alpha_s$) for a given confidence level. Adopting a *normalized* non-conformity measure instead provides personalized individual bounds for each new example by scaling the standard non-conformity measure with σ_i, a term that estimates the difficulty of predicting y_i. This means that using a *normalized* non-conformity measure gives a smaller prediction interval for "easy" examples, and a bigger one for "hard" examples. Thus, two distinct examples with the same α_s calculated by (2) will have two different interval predictions depending on their difficulty. In this case, the normalized non-conformity score is as follows:

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\sigma_i}. \tag{5}$$

Thus, we have:

$$\mathbb{P}\left(\frac{|y_i - \hat{y}_i|}{\sigma_i} \leq \alpha_s\right) \geq 1 - \epsilon, \tag{6}$$

which becomes an equality if the method is perfectly calibrated. For a new example x_{n+1}, the prediction interval becomes :

$$\hat{\mathbf{y}}_{n+1} = \left[\hat{y}_{n+1} - \alpha_s\sigma_{n+1}, \hat{y}_{n+1} + \alpha_s\sigma_{n+1}\right]. \tag{7}$$

The value σ_i can be defined in various ways. A popular approach proposed by Papadopoulos and Haralambous [14] consists of training a small neural network to estimate the error of the underlying algorithm by predicting the value $\mu_i = \ln(|y_i - \hat{y}_i|)$. In this case, the non-conformity score is defined as:

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\exp(\mu_i) + \beta}, \tag{8}$$

where $\beta \geq 0$ is a sensitivity parameter. With the significance level ϵ, we have:

$$\mathbb{P}\left(\frac{|y_i - \hat{y}_i|}{\exp(\mu_i) + \beta} \leq \alpha_s\right) \geq 1 - \epsilon. \tag{9}$$

For a new example x_{n+1}, the prediction interval is:

$$\hat{\mathbf{y}}_{n+1} = \left[\hat{y}_{n+1} - \alpha_s(\exp(\mu_{n+1}) + \beta), \hat{y}_{n+1} + \alpha_s(\exp(\mu_{n+1}) + \beta)\right]. \tag{10}$$

Other approaches use different algorithms to normalize the non-conformity scores, such as regression trees [15] and k-nearest neighbors [10]. Before introducing the problem of multi-target regression, let us first note that, assuming

that our method is well-calibrated and that $|y_i - \hat{y}_i|/\sigma_i$ is associated to a random variable Q, (6) can be rewritten as

$$\mathbb{P}(Q \leq \alpha_s) = 1 - \epsilon := F_Q(\alpha_s), \tag{11}$$

which will be instrumental when dealing with copulas and multi-variate outputs later on. Also note that this means that specifying a confidence ϵ uniquely defines a value α_s.

2.2. Multi-target regression (MTR)

In multi-target regression, the feature space X is the same as in standard regression, but the target space $Y \subset \mathbb{R}^m$ is made of m real-valued targets. This means that observations are i.i.d pairs (x_i, y_i) drawn from a probability distribution on $X \times Y$, where each instance $x_i \in X$ is associated to an m dimensional real-valued target $y_i = (y_i^1, \ldots, y_i^m) \in Y$. The usual objective of multi-target regression is then to learn a predictor $h : X \to Y$, i.e. to predict multiple outputs based on the input features characterizing the data set, which generalizes standard regression. There are two distinct approaches to treat MTR called *algorithm adaptation* and *problem transformation* methods.

For *algorithm adaptation* approaches, standard single-output regression algorithms are extended to the multi-target regression problem. Many models were adapted to the MTR problem, such as Support Vector Regressors [16], regression trees [17], kernel methods [18] and rule ensembles [19].

In *problem transformation*, one usually decomposes the initial multi-variate problems into several simpler problems, thus allowing the use of standard classification methods without the need for an adaptation that can be tricky or computationally costly. A prototypical example of such a transformation is the chaining method [20], where one predicts each target sequentially, using the output and predictions of previous targets as inputs for the next one, thus capturing some correlations between the targets.

As our goal here is not to produce a new MTR method, but rather to propose a flexible means to make their predictions reliable through conformal prediction,

we will not make a more detailed review of those methods. The reader interested in different methods can consult for instance [20]. We will now detail how conformal prediction and MTR can be combined. Let us just mention that exploiting the possible relationships allow in general to improve performances of the methods [21, 22].

2.3. Inductive conformal prediction for Multi-Target Regression

As said before, previous studies about conformal MTR focused on providing valid and efficient inferences target-wise [9], thus potentially neglecting the potential advantages of exploiting target relations. Our main goal in this paper is to provide an easy conformal MTR method allowing to do so.

Within the MTR setting, we have a multi-dimensional output $\{Y^1, \ldots, Y^m\}$ (we will use superscripts to denote the dimensions, and subscripts to denote sample indices) with $Y^j \in \mathbb{R}, j \in \{1, \ldots, m\}$ the different individual real-valued m targets. Let $\underline{\hat{y}}^j_{n+1}, \overline{\hat{y}}^j_{n+1}$ be respectively the lower and upper bounds of the interval predictions given by the non-conformity measure for each target Y^j given a new instance x_{n+1}. We define the hyper-rectangle $[\hat{\mathbf{y}}_{n+1}]$ as the following Cartesian product:

$$[\hat{\mathbf{y}}_{n+1}] = \times_{j=0}^{m} [\underline{\hat{y}}^j_{n+1}, \overline{\hat{y}}^j_{n+1}]. \tag{12}$$

This hyper-rectangle forms the volume $\prod_{j=0}^{m} (\overline{\hat{y}}^j_{n+1} - \underline{\hat{y}}^j_{n+1})$ to which a global prediction y_{n+1} of a new example x_{n+1} should belong in order to be valid, i.e. each single prediction y^j_{n+1} for each individual target Y^j should be between the bounds $\underline{\hat{y}}^j_{n+1}, \overline{\hat{y}}^j_{n+1}$ of its interval prediction. With this view, the objective of the conformal prediction framework for MTR in the normalized setting is to satisfy a global significance level ϵ_g required by the user such that:

$$\mathbb{P}(y_{n+1} \in [\hat{\mathbf{y}}_{n+1}]) \geq 1 - \epsilon_g. \tag{13}$$

This probability can also be written as follows:

$$\begin{aligned} &\mathbb{P}(y^1_{n+1} \in [\underline{y^1_{n+1}}, \overline{y^1_{n+1}}], \ldots, y^m_{n+1} \in [\underline{y^m_{n+1}}, \overline{y^m_{n+1}}]) \\ &= \mathbb{P}\left(\frac{|y^1_{n+1} - \hat{y}^1_{n+1}|}{\sigma^1_{n+1}} \leq \alpha^1_s, \ldots, \frac{|y^m_{n+1} - \hat{y}^m_{n+1}|}{\sigma^m_{n+1}} \leq \alpha^m_s \right) \geq 1 - \epsilon_g. \end{aligned} \tag{14}$$

Thus, we need to find the individual non-conformity scores $\alpha_s^1, \ldots, \alpha_s^m$, defined for instance by target-wise confidence levels ϵ_j, such that we ensure a global confidence level $1 - \epsilon_g$. Extending (11) and considering the random variables $Q^j = |y^j - \hat{y}^j|/\sigma^j$, $j \in \{1, \ldots, m\}$, we get:

$$\mathbb{P}(Q^1 \leq \alpha_s^1, \ldots, Q^m \leq \alpha_s^m) \geq 1 - \epsilon_g. \tag{15}$$

Should we know the joint distribution in (15), and therefore the dependence relations between target predictions, it would be relatively easy to get the individual significance levels[1] ϵ_j associated to the individual non-conformity scores α_s^j such that we satisfy the chosen confidence level $1-\epsilon_g$. Yet, such a joint distribution is usually unknown. The next section proposes a simple and efficient method to do so, leveraging the connection between (15) and copulas. Before doing that, note again that under the assumption that we are well calibrated, we can transform (15) into

$$F(\alpha_s^1, \ldots, \alpha_s^m) = 1 - \epsilon_g, \tag{16}$$

where F denotes here the joint cumulative distribution induced by $\mathbb{P}$.

3. Copula-based conformal Multi-Target Regression

This section introduces our approach to obtain valid or better conformal prediction in the multi-variate regression setting. We first recall some basics of copulas and refer to Nelsen [12] for a full introduction, before detailing how we apply them to conformal approaches.

3.1. Overview on copulas

A copula is a mathematical function that can describe the dependence between multiple random variables. The term "copula" was first introduced by Sklar [23] in his famous theorem, which is one of the fundamentals of copula

[1] Note that there may be multiple choices for such individual levels. Here we will fix them to be equal for simplicity.

theory, now known as Sklar's theorem. However, these tools have already been used before, as for instance in Fréchet's paper [24] and Höffding's work [25, 26] (reprinted as [27]). Copulas are popular in the statistical and financial fields [28], but they are nowadays more and more used in other domains as well, such as hydrology [29], medicine [30], and machine learning [31].

Let $\mathbf{Q} = (Q^1, \ldots, Q^m)$ be an m-dimensional random vector composed of the random variables $Q^1, \ldots, Q^m$. Let its cumulative distribution function (c.d.f.) be $F = F_Q : \mathbb{R}^m \to [0, 1]$. This c.d.f. carries two important pieces of information:

- The c.d.f. of each random variable Q^j s.t. $F_j(q^j) = \mathbb{P}(Q^j \leq q^j)$, for all $j \in \{1, \ldots m\}$.
- The dependence structure between them.

The objective of copulas is to isolate the dependence structure from the marginals Q^j by transforming them into uniformly distributed random variables U^j and then expressing the dependence structure between the U^j's. In other words, an m-dimensional copula $C : [0, 1]^m \to [0, 1]$ is a c.d.f. with standard uniform marginals. It is characterized by the following properties:

1. C is grounded, i.e. if $u^j = 0$ for at least one $j \in \{1, \ldots, m\}$, then $C(u^1, \ldots, u^m) = 0$.
2. If all components of C are equal to 1 except u^j for all $u^j \in [0, 1]$ and $j \in \{1, \ldots, m\}$, then $C(1, \ldots, 1, u^j, 1, \ldots, 1) = u^j$.
3. C is m-increasing, i.e., for all $\mathbf{a}, \mathbf{b} \in [0, 1]^m$ with $\mathbf{a} \leq \mathbf{b}$:

$$\Delta_{(\mathbf{a},\mathbf{b}]} C = \sum_{j \in \{0,1\}^m} (-1)^{\sum_{k=1}^m j_k} C(a_1^{j_1} b_1^{1-j_1}, \ldots, a_m^{j_m} b_m^{1-j_m}) \geq 0.$$

The last inequality simply ensures that the copula is a well-defined c.d.f. inducing non-negative probability for every event. The idea of copulas is based on probability and quantile transformations [32]. Using these latter, we can see that all multivariate distribution functions include copulas and that we can use a mixture of univariate marginal distributions and a suitable copula to produce

a multivariate distribution function. This is described in Sklar's theorem [23] as follows:

Theorem 3.1 (Sklar's theorem). *For any m-dimensional cumulative distribution function (c.d.f.) F with marginal distributions $F_1, \ldots, F_m$, there exists a copula $C : [0,1]^m \to [0,1]$ such that:*

$$F(\mathbf{q}) = F(q^1, \ldots, q^m) = C(F_1(q^1), \ldots, F_m(q^m)), \quad \mathbf{q} \in \mathbb{R}^m. \tag{17}$$

If F_j is continuous for all $j \in \{1, \ldots, m\}$, then C is unique.

Denoting the pseudo inverse of F_j as $F_j^{\leftarrow}$ [32], we can get from (17) that

$$C(\mathbf{u}) = C(u^1, \ldots, u^m) = F(F_1^{\leftarrow}(u^1), \ldots, F_m^{\leftarrow}(u^m)). \tag{18}$$

There are a few noticeable copulas, among which are:

- the product copula: $\Pi(\mathbf{u}) = \prod_{j=1}^{m} u^j$;
- the Fréchet-Höffding upper bound copula [2]: $M(\mathbf{u}) = \min_{1 \le j \le m}\{u^j\}$;
- the Fréchet-Höffding lower bound copula [3]: $W(\mathbf{u}) = \max\{\sum_{j=1}^{m} u^j - m + 1, 0\}$.

While the product copula corresponds to classical stochastic independence, the Fréchet-Höffding bound copulas play an important role as they correspond to extreme cases of dependence [33]. Indeed, any m-dimensional copula C is such that $W(\mathbf{u}) \le C(\mathbf{u}) \le M(\mathbf{u}), \mathbf{u} \in [0,1]^m$.

Another important class of copulas are so-called Archimedean copulas, which are based on generator functions ϕ of specific kinds. More precisely, a continuous, strictly decreasing, convex function $\phi : [0,1] \to [0,\infty]$ satisfying $\phi(1) = 0$ is known as an Archimedean copula generator. It is known as a strict generator if $\phi(0) = \infty$. The generated copula is then given by

$$C(u^1, \ldots, u^m) = \phi^{[-1]}(\phi(u^1) + \ldots + \phi(u^m)). \tag{19}$$

[2] M is a copula for all $m \ge 2$.
[3] W is a copula if and only if $m = 2$.

Table 1 provides examples and details of three one parameter Archimedean copula families [32], which are particularly convenient in estimation problems (being based on a single parameter).

Family	Generator $\phi(t)$	θ range	Strict	Lower	Upper
Gumbel [34]	$(-\ln t)^\theta$	$\theta \geq 1$	Yes	Π	M
Clayton [35]	$\frac{1}{\theta}(t^{-\theta}-1)$	$\theta \geq -1$	$\theta \geq 0$	W	M
Frank [36]	$-\ln\left(\frac{e^{-\theta t}-1}{e^{-\theta}-1}\right)$	$\theta \in \mathbb{R}$	Yes	W	M

Table 1: Archimedean copula families.

3.2. Copula-based conformal Multi-Target Regression

Let us now revisit our previous problem of finding the significance levels ϵ_j for each target so that the hyper-rectangle prediction $[\hat{\mathbf{y}}]$ covers the true value with confidence $1-\epsilon_g$. Let us first consider (16). Following Sklar's theorem, we have

$$\begin{aligned} F(\alpha_s^1,\ldots,\alpha_s^m) &= C(F_1(\alpha_s^1),\ldots,F_m(\alpha_s^m)) \\ &= C(1-\epsilon^1,\ldots,1-\epsilon^m) \\ &= 1-\epsilon_g \end{aligned}$$

where the second line is obtained from (6). Clearly, if we knew the copula C, then we could search for values ϵ_j providing the desired global confidence.

A major issue is then to obtain or estimate the copula modelling the dependence structure between the targets and their confidence levels. As copulas are classically estimated from multi-variate observations, a simple means that we will use here is to estimate them from the non-conformity scores generated from the calibration set Z^{cal}. Namely, if α_i^j is the non-conformity score corresponding to the j^{th} target of the z_i example of Z^{cal} for $i \in \{l+1,\ldots,n\}$, we simply

propose to estimate a copula C from the matrix

$$A = \begin{bmatrix} \alpha_{l+1}^1 & \alpha_{l+1}^2 & \dots \\ \vdots & \ddots & \\ \alpha_n^1 & & \alpha_n^m \end{bmatrix}. \tag{20}$$

3.3. On three specific copulas

We will now provide some detail about the copulas we performed experiments on. They have been chosen to go from the one requiring the most assumptions to the one requiring the least assumptions.

3.3.1. The Independent copula

The Independent copula means that the m targets are considered as being independent, with no relationship between them. It is a strong assumption, but it does not require any estimation of the copula. In this case, (15) becomes:

$$\begin{aligned} \Pi(F_1(\alpha_s^1), \dots, F_m(\alpha_s^m)) &= \prod_{j=1}^{m} F_j(\alpha_s^j) = \prod_{j=1}^{m} \mathbb{P}(Q^j \leq \alpha_s^j) \\ &\geq \prod_{j=1}^{m} (1 - \epsilon^j) = 1 - \epsilon_g, \end{aligned}$$

If we assume that all $\epsilon^1, \dots, \epsilon^m$ equal the same value ϵ_t, then:

$$\prod_{j=1}^{m} (1 - \epsilon^j) = (1 - \epsilon_t)^m = 1 - \epsilon_g.$$

Thus, we simply obtain

$$\epsilon_t = 1 - \sqrt[m]{1 - \epsilon_g}. \tag{21}$$

This individual significance level ϵ_t is then used to calculate the different non-conformity scores α_s^j for each target in the multi-target regression problem for the Independent copula.

3.3.2. The Gumbel copula

The Gumbel copula is a member of the Archimedean copula family which depends on only one parameter, and in this sense is a good representative

of parametric copulas. It comes down to applying the generator function $\phi(F_j(\alpha_s^j)) = (-\ln F_j(\alpha_s^j))^\theta$ and its inverse $\phi^{[-1]}(F_j(\alpha_s^j)) = \exp -(F_j(\alpha_s^j))^{1/\theta}$ to (19), resulting in the expression

$$C_G^\theta(F_1(\alpha_s^1), \ldots, F_m(\alpha_s^m)) = \exp - \left(\sum_{j=1}^{m} \left(-\ln F_j(\alpha_s^j) \right)^\theta \right)^{1/\theta}. \tag{22}$$

In this case, we need to estimate the parameter θ. Since the marginals $F_j(\alpha^j)$ are unknown, we also need to estimate them. In our case, we will simply use the empirical c.d.f. induced by the non-conformity scores α_i^j of matrix A. An alternative would be to also assume a parametric form of the F_j, but this seems in contradiction with the very spirit of non-conformity scores. In particular, we will denote by $\hat{F}_j$ the empirical cumulative distribution such that

$$\hat{F}_j(\beta) = \frac{|\{\alpha_i^j : \alpha_i^j \leq \beta, i \in \{l+1, \ldots, n\}\}|}{n-l}, \quad \beta \in \mathbb{R}.$$

The parameter θ can then be estimated from matrix A using the Maximum Pseudo-Likelihood Estimator [37] with a numerical optimization, for instance by using the Python library "copulae"[4]. Once this is obtained, we then get for a particular choice of ϵ_j that

$$C_G^{\hat{\theta}} = \exp - \left(\sum_{j=1}^{m} \left(-\ln(1-\epsilon_j) \right)^{\hat{\theta}} \right)^{1/\hat{\theta}} \tag{23}$$

$$= \exp - \left(\sum_{j=1}^{m} \left(-\ln F_j(\alpha_s^j) \right)^{\hat{\theta}} \right)^{1/\hat{\theta}} \tag{24}$$

And we can search for values ϵ_j that will make this equation equal to $1-\epsilon_g$, using the estimations $\hat{F}_j$. The solution is especially easy to obtain analytically if we consider that $\epsilon^1 = \ldots = \epsilon^m = \epsilon_t$, as we then have that

$$\epsilon_t = 1 - (1-\epsilon_g)^{1/\sqrt[\theta]{m}},$$

and one can then obtain the corresponding non-conformity scores $\alpha_s^1, \ldots, \alpha_s^m$ by replacing F_j by $\hat{F}_j$.

[4] https://pypi.org/project/copulae/

We chose this particular family of Archimedean copulas because its lower bound is the Independent copula (as seen in Table 1). We can easily verify this by taking $\hat{\theta} = 1$. Thus, we can capture independence if it is verified, and otherwise search in the direction of positive dependence. One reason for such a choice is that previous experiments [11] indicate that the product copula gives overly conservative results.

3.3.3. The Empirical copula

Parametric copulas, as all parametric models, have the advantage of requiring less data to be well estimated, while having the possibly important disadvantage that they induce some bias in the estimation, that is likely to grow as the number of target increases. The Empirical copula presents a non-parametric way of estimating the marginals directly from the observations [38, 39]. It is defined as follows [37]:

$$C_E(\mathbf{u}) = \frac{1}{n-l} \sum_{i=l+1}^{n} \mathbb{1}_{\mathbf{u}_i \leq \mathbf{u}} = \frac{1}{n-l} \sum_{i=l+1}^{n} \prod_{j=1}^{m} \mathbb{1}_{u_i^j \leq u^j}, \quad \mathbf{u} \in [0,1]^m, \tag{25}$$

where $\mathbb{1}_A$ is the indicator function of event A, and the inequalities $\mathbf{u}_i \leq \mathbf{u}$ for $i \in \{l+1, \ldots, n\}$ need to be understood component-wise. $\mathbf{u}_i$ are the pseudo-observations that replace the unknown marginal distributions, which are defined as:

$$\mathbf{u}_i = (u_i^1, \ldots, u_i^m) = (\hat{F}_1(\alpha_i^1), \ldots, \hat{F}_m(\alpha_i^m)), \quad i \in \{l+1, \ldots, n\}, \tag{26}$$

where distributions $\hat{F}_j$ are defined as before. Simply put, the Empirical copula corresponds to consider as our joint probability the Empirical joint cumulative distribution. We then have that

$$C_E(F_1(\alpha_s^1), \ldots, F_m(\alpha_s^m)) = \frac{1}{n-l} \sum_{i=l+1}^{n} \prod_{j=1}^{m} \mathbb{1}_{u_i^j \leq F_j(\alpha_s^j)}. \tag{27}$$

Using that $F_j(\alpha_s^j) = 1 - \epsilon_j$, we can then search for values of ϵ_j, $j = 1, \ldots, m$ that will make (27) equal to $1 - \epsilon_g$. Note that in this case, even assuming that $\epsilon^1 = \ldots = \epsilon^m = \epsilon_t$ will require an algorithmic search, which is however easy as C_E is an increasing function, meaning that we can use a simple dichotomic search.

4. Evaluation

In this section, we describe the experimental setting (underlying algorithm, data sets and performance metrics) and the results of our study.

4.1. Experimental setting

We choose to work with a deep Neural Network (NN) and a Random Forest (RF) as the underlying algorithms, and compare between the three copula functions to show that adding copulas to the non-conformity measures works with any underlying algorithm. However, our approach can be easily adapted to any multi-variate regression model.

To compute the non-conformity scores over the calibration set, we use the normalized non-conformity score given by (8) as described in [14], and predict $\mu_i = \ln(|y_i - \hat{y}_i|)$ simultaneously for all targets by a single multivariate multi-layer perceptron. In this case, μ_i represents the estimation of the underlying algorithm's error. As mentioned before, the approach can be adapted to any conformal regression approach.

Experiments are conducted on normalized data with a mean of 0 and a standard deviation of 1, with a 10-fold cross validation to avoid the impact of biased results, and with a calibration set equal to 10% of the training examples for all data sets. We take the value $\beta = 0.1$ for the sensitivity parameter and do not optimize it when calculating the normalizing coefficient μ_i. After getting the proper training data (X^{tr}, Y^{tr}), calibration data (X^{cal}, Y^{cal}) and test data (X^{ts}, Y^{ts}) for each fold, we follow the steps described below:

1. Train the underlying algorithm (NN or RF) on the proper training data (X^{tr}, Y^{tr}). The Neural Network's architecture is composed of a first dense layer applied to the input with "selu" activation (scaled exponential linear units [40]), three hidden dense layers with dropouts and "selu" activation, and a final dense layer with m outputs and a linear activation. The Random Forest is trained for each target alone using Python sklearn's implementation, then each target is predicted independently to get the results.

2. Predict $\hat{Y}^{cal}$ and $\hat{Y}^{ts}$ for calibration and test data respectively using the underlying algorithm.
3. Train the normalizing multi-layer perceptron on the proper training data $(X^{tr}, \mu_{tr} = \ln(|Y^{tr} - \hat{Y}^{tr}|)$, corresponding to the error estimation of the underlying algorithm. The normalizing MLP consists of three hidden dense layers with "selu" activation and dropouts and a final dense layer with m outputs for predicting all targets simultaneously. This approach was chosen since it proved to be more effective than a single target approach that we experimented in a previous work [11].
4. Predict μ_{cal} and μ_{ts} for calibration and test data respectively using the normalizing MLP.
5. If needed, get an estimation[5] of the copula C from the matrix A of calibration non-conformity scores.
6. For each global significance level ϵ_g:
   - Get the individual significance level $\epsilon_j = \epsilon_t$ for $j \in \{1, \ldots, m\}$ and calculate $\alpha_s = \{\alpha_s^1, \ldots, \alpha_s^m\}$ for all targets using calibration data, according to the methods mentioned in Section 3.3.
   - Get the interval predictions for the test data with:

$$\left[\hat{Y}^{ts} - \alpha_s(\exp(\mu_{ts}) + \beta), \hat{Y}^{ts} + \alpha_s(\exp(\mu_{ts}) + \beta)\right]. \tag{28}$$

Remark 4.1. *We choose $\epsilon_j = \epsilon_t$ for $j \in \{1, \ldots, m\}$ as we have no indication that individual targets should be treated with different degree of cautiousness. However, since copulas are functions from $[0,1]^m$ to $[0,1]$, there is in principle no problem in considering different confidence degrees for different tasks, if an application calls for it. How to determine and elicit such degrees is however, to our knowledge, an open question.*

The implementation was done using Python and Tensorflow. The copula part

[5] In the case of the Gumbel copula, we use a Maximum Pseudo-Likelihood Estimator with a numerical optimization using the BFGS algorithm

Names	Examples	Features	Targets
music origin [41]	1059	68	2
indoor loc [42]	21049	520	3
scpf [43]	1137	23	3
sgemm [44]	241600	14	4
rf1 [43]	9125	64	8
rf2 [43]	9125	576	8
scm1d [43]	9803	280	16
scm20d [43]	8966	61	16

Table 2: Information on the used multi-target regression data sets.

of our experiments was based on the book [37] and the Python library "copulae". The code used for this paper is available in Github[6].

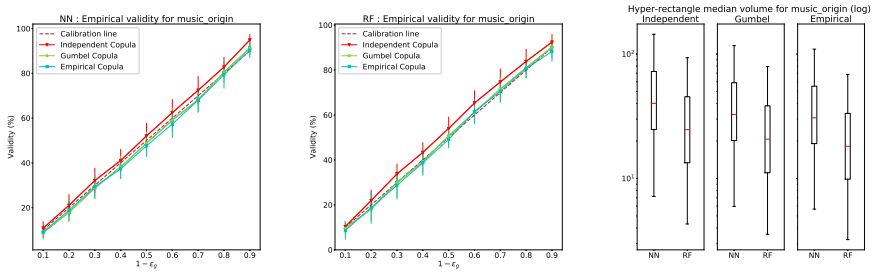
We use eight data sets with different numbers of targets and varying sizes. They are summarized in Table 2.

4.2. Results

This section presents the results of our experiments, investigating in particular the validity and efficiency of the proposed approaches. Figures 1 and 2 detail these results for "music origin" and "sgemm". The figures for all other data sets can be found in Appendix A.

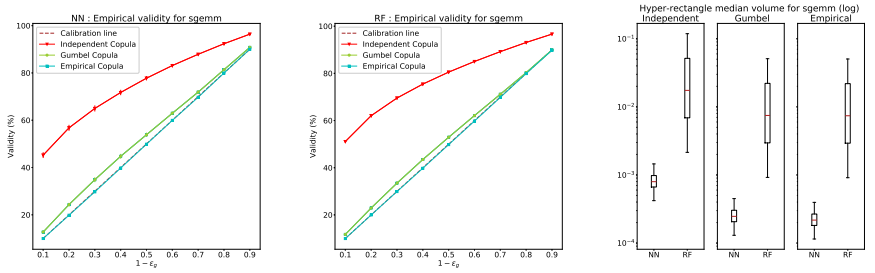
To verify the validity of each non-conformity measure, we calculate the accuracy of each one and compare it with the calibration line. This line represents the case where the error rate is exactly equal to ϵ_g for a confidence level $1 - \epsilon_g$, which is the desired outcome of using conformal prediction. In multi-target regression, the accuracy is computed based on whether the observation y belongs to the hyper-rectangle $[\hat{\mathbf{y}}]$ or not depending on the significance level ϵ_g. Thus, a

[6] https://github.com/M-Soundouss/CopulaConformalMTR



(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure 1: Results for music origin.



(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure 2: Results for sgemm.

correctly predicted example must verify that all its individual predictions y_i for each individual target Y_i is in its corresponding individual interval predictions. Concretely, for each considered confidence level ϵ_g and test example $x \in X^{ts}$, we obtain a prediction $[\hat{\mathbf{y}}]_{\epsilon_g}$. From this, we can compute the empirical validity as the percentage of times that $[\hat{\mathbf{y}}]_{\epsilon_g}$ contains the true observed value, i.e.,

$$\frac{\sum_{(x,y)\in Z^{ts}} \mathbb{1}_{y\in[\hat{\mathbf{y}}]_{\epsilon_g}}}{|Z^{ts}|}.$$

Doing it for several values of ϵ_g, we obtain a calibration curve that should be as close as possible to the identity function.

The results of the error rate or accuracy curves are shown in sub-figures (a) for the Neural Network and (b) for the Random Forest of each Figure 1 and 2. The curves correspond to the Independent, Gumbel and Empirical multivariate

non-conformity measures. The results clearly show that the best performance is obtained by using the Empirical copula, where the model is well calibrated. For most of the studied data sets, the Empirical copula accuracy curve is almost perfectly aligned with the calibration line, and thus almost exactly valid. This is due to the fact that Empirical copula functions use non-parametric estimate of the marginals based on the observations, which enables the model to better adapt to the dependence structure of each data set. This dependence structure is neglected when using an Independent copula-based non-conformity measure, since the m targets are treated as if they were independent, and so the link between them is not exploited when computing ϵ_t. This also means that the difference between the Empirical and the Independent copula-based non-conformity measures is bigger when there is a strong dependence between the non-conformity scores, and is an indication of the strength of this dependence. For instance, we can deduce that the targets are strongly related for "sgemm" by the big gap between the Independent and Empirical accuracy curves (Figures 2a and 2b). For the Gumbel copula, the accuracy curve is generally closer to the calibration line than the one for the Independent copula. This supports the existence of a dependence structure between the targets, since the lower bound of the Gumbel copula is the Independent copula, which means that if the targets were in fact independent, the two curves would perfectly match. This can be seen in Figures 1a and 1b for "music origin", where the accuracy curves almost overlap all the time, meaning that the targets are likely to be independent. These conclusions concerning the empirical efficiency are the same for both underlying algorithms, which suggests that the difference regarding the validity performance mainly comes from the chosen copula-based non-conformity measure.

From the empirical validity results, we also noticed that the Empirical copula non-conformity measure can be slightly invalid sometimes (Figures A.6a and A.6b for "scpf"). We explain this by the fewer number of examples, in which case one could use a more regularized form than the Empirical copula. However, when a lot of examples are available (for instance, more than 200000 observations for "sgemm"), the validity curve of the Empirical copula non-conformity measure is

perfectly aligned with the calibration line, meaning that this measure is exactly valid (Figures 2a and 2b).

In single-output regression, efficiency is measured by the size of the intervals, and a method is all the more efficient as predicted intervals are small. To assess efficiency in multi-target regression, we can simply compute the volume of the obtained predictions $[\hat{\mathbf{y}}]_{\epsilon_g}$, after (12). For each experiment, we then compute the median value of those hyper-rectangle volumes (for the estimation to be robust against very large hyper-rectangles).

Efficiency results are shown in sub-figure c for all data sets for $\epsilon_g = 0.1$. They show that, for each underlying algorithm, the Independent copula has a bigger median hyper-rectangle volume compared to the Gumbel and Empirical copulas, especially in those cases where the existence of a dependence structure is confirmed by the calibration curves. This is due to the fact that using an Independent copula ignores the dependence between the non-conformity scores, which leads to an over-estimation of the global hyper-rectangle error. This impact is avoided when using the Empirical copula because it takes advantage of the dependence structure to construct better interval predictions. Another remark concerning efficiency is that the box plots for Empirical copula are tighter than the other two, which shows that the values are homogeneous on all folds compared to the Independent copula for instance, where the variation is much more visible. When comparing between the underlying algorithms, we can see that the Neural Network gives tighter volumes for "sgemm" (Figure 2c), whereas the Random Forest gives better results for "music origin" (Figure 1c). We can explain this by the fact that "sgemm" has more data, and the strong dependence structure is taken into consideration when training the Neural Network that is trained on all targets simultaneously, as opposed to the Random Forest that is trained on each target individually.

The empirical validity and hyper-rectangle median volume results are summarized in Tables 3 and 4. The validity simply provides the average difference between a perfect calibration (the identity function) and the observed curve for each copula. This means, in particular, that a negative value indicates that the

	Independent		Gumbel		Empirical	
	NN	RF	NN	RF	NN	RF
music origin	2.19 ± 4.89	3.32 ± 4.68	-0.93 ± 4.66	$\mathbf{0.17 \pm 4.93}$	-1.41 ± 4.84	-0.56 ± 5.14
indoor loc	3.77 ± 1.11	3.89 ± 1.56	1.8 ± 1.16	1.09 ± 1.39	$\mathbf{0.03 \pm 1.13}$	0.12 ± 1.4
scpf	22.33 ± 4.79	18.56 ± 4.32	15.6 ± 4.7	11.57 ± 5.01	-3.47 ± 4.87	$\mathbf{0.48 \pm 5.79}$
sgemm	25.14 ± 0.84	28.07 ± 0.4	3.06 ± 0.68	1.99 ± 0.39	$\mathbf{-0.14 \pm 0.39}$	-0.15 ± 0.39
rf1	6.01 ± 1.44	4.99 ± 1.28	2.98 ± 1.38	1.98 ± 1.33	-0.4 ± 1.49	$\mathbf{-0.34 \pm 1.48}$
rf2	5.78 ± 2.68	4.94 ± 1.76	3.08 ± 2.37	1.98 ± 1.89	-0.3 ± 1.6	$\mathbf{0.24 \pm 1.68}$
scm1d	14.77 ± 2.84	14.58 ± 2.89	10.66 ± 2.67	9.79 ± 2.84	$\mathbf{-0.57 \pm 1.85}$	-0.79 ± 2.3
scm20d	14.44 ± 2.06	14.97 ± 2.02	10.52 ± 2.33	9.39 ± 2.1	$\mathbf{-1.16 \pm 2.01}$	-1.54 ± 2.09

Table 3: Validity (average gap between the empirical validity curve and the calibration line in percentage) summarized results for all data sets.

observed frequency is in average below the specified confidence degree.

The numbers confirm our previous observations on the graphs, as the average gap is systematically higher for the Independent copula and lower for the Empirical one, with Gumbel in-between. We can however notice that while the Empirical copula provides the best results, it is also often a bit under the calibration line, indicating that if conservativeness is to be sought, one should maybe prefer the Gumbel copula. These outcomes are the same for both NN and RF, without one algorithm being overall better than the other. About the same conclusions can be given regarding efficiency, with the Empirical copula giving the best results and the Independent one the worst.

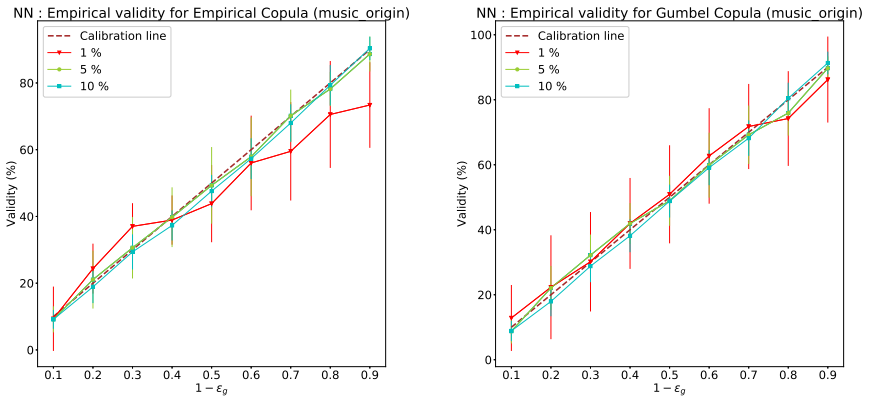
To complete our experiments and analyze the sensitivity of our approach to the size of the calibration set, we conducted the same experiments on two datasets, where we retained only 1% and 5% of the whole data set: "indoor loc" which has a lot of examples (21049) and "music origin" which has fewer examples (1059). We only used Neural Networks as the underlying algorithm with the Empirical and Gumbel copulas non-conformity measures to compare between them. Figures 3 and 4 show the results for both datasets.

Results clearly show that with fewer examples for "music origin", the Em-

	Independent		Gumbel		Empirical	
	NN	RF	NN	RF	NN	RF
music origin	$4.02^{1} \pm 1.54^{1}$	$2.47^{1} \pm 1.18^{1}$	$3.27^{1} \pm 1.5^{1}$	$2.07^{1} \pm 1.1^{1}$	$3.08^{1} \pm 1.46^{1}$	$\mathbf{1.81^{1} \pm 7.82}$
indoor loc	$1.31^{-1} \pm 7.77^{-2}$	$4.76^{-1} \pm 7^{-1}$	$1.15^{-1} \pm 7.95^{-2}$	$4.13^{-1} \pm 6.02^{-1}$	$\mathbf{1.03^{-1} \pm 7.65^{-2}}$	$4.26^{-1} \pm 6.5^{-1}$
scpf	$1.03^{11} \pm 3.02^{11}$	$8.72^{10} \pm 2.06^{11}$	$1.02^{11} \pm 3.02^{11}$	$7.56^{10} \pm 2.08^{11}$	$1.12^{7} \pm 2.05^{7}$	$\mathbf{5.71^{6} \pm 1.54^{7}}$
sgemm	$7.97^{-4} \pm 4.81^{-4}$	$1.75^{-2} \pm 2.58^{-3}$	$2.47{-}4 \pm 1.45^{e-4}$	$7.48^{-3} \pm 7.91^{-4}$	$\mathbf{2.17^{-4} \pm 1.25^{-4}}$	$7.4^{-3} \pm 8.15^{-4}$
rf1	$7.19^{-3} \pm 1.23^{-2}$	$5.64^{-5} \pm 4.87^{-5}$	$5.15^{-3} \pm 9.11^{-3}$	$3.56^{-5} \pm 3.44^{-5}$	$4.49^{-3} \pm 9.23^{-3}$	$\mathbf{2.81^{-5} \pm 1.67^{-5}}$
rf2	$2.17^{-3} \pm 2.89^{-3}$	$2.67^{-4} \pm 3.54^{-4}$	$1.67^{-3} \pm 2.45^{-3}$	$\mathbf{1.42^{-4} \pm 1.71^{-4}}$	$1.52^{-3} \pm 2.42^{-3}$	$\mathbf{1.42^{-4} \pm 1.71^{-4}}$
scm1d	$1.08^{5} \pm 1.04^{5}$	$1.72^{5} \pm 1.66^{5}$	$1.67^{4} \pm 1.33^{4}$	$2.11^{4} \pm 1.58^{4}$	$\mathbf{2.31^{3} \pm 1.93^{3}}$	$3.43^{3} \pm 2.27^{3}$
scm20d	$2.18^{6} \pm 4.26^{6}$	$5.77^{6} \pm 5.38^{6}$	$2.14^{5} \pm 2.88^{5}$	$1.02^{6} \pm 7.3^{5}$	$\mathbf{2.73^{4} \pm 2.58^{4}}$	$2.01^{5} \pm 1.06^{5}$

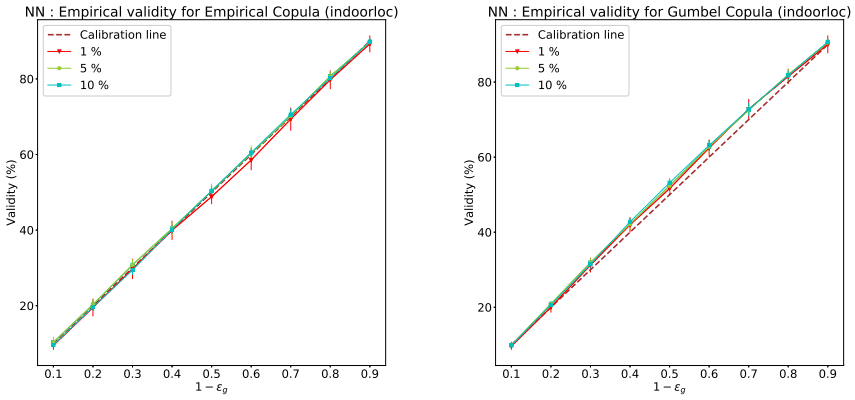
We note X^{Y} the value $X \times 10^{Y}$.

Table 4: Efficiency (hyper-rectangle median volume for $\epsilon_g = 0.1$) summarized results for all data sets.



(a) Validity for the Empirical copula (b) Validity for the Gumbel copula

Figure 3: Results for music origin for different calibration data sizes.



(a) Validity for the Empirical copula (b) Validity for the Gumbel copula

Figure 4: Results for indoor loc for different calibration data sizes.

pirical non-conformity measure is often invalid and also unstable (has larger variability) with 1% of the examples as calibration data (Figure 3a). This can also be seen in the difference of variance between values for the empirical validity, with 10% having more homogeneous values as compared to 5% and 1% respectively. Using the Gumbel copula, which is semi-parametric, helps to attenuate the effect, with more consistent results even for 1% (Figure 3b). For "indoor loc", the impact of the percentage of data used is insignificant, since the validity curves overlap for 10%, 5% and 1% of data used for calibration, mainly because 1% of the whole data set is still quite large (about 200 samples, to be compared with the 10 samples of "music origin"). This is the case for both Empirical and Gumbel copulas, giving the same results as earlier in Figure A.5a, i.e. the Empirical copula being exactly valid and better than the Gumbel copula.

5. Conclusion and discussion

In this paper, we provided a quite easy and flexible way to obtain valid conformal predictions in a multi-variate regression setting. We did so by exploiting a link between non-conformity scores and copulas, a commonly used tool to

model multi-variate distribution.

Experiments on various data sets for a small choice of representative copulas show that the method indeed allows to improve upon the naive independence assumption for different underlying algorithms (Neural Networks and Random Forests). Those first results indicate in particular that while parametric, simple copulas may provide valid results for some data sets, more complex copulas may be needed in general to obtain well calibrated predictions, with the cost that good estimations of such copulas require a lot of calibration data.

As future lines of work, we would like to explore further the flexibility of our framework, for instance by adapting it to the richer conformal predictive distributions [45], by exploring the possibility of using vines [46] to model complex dependencies, or by proposing protocols allowing to obtain ϵ_g from different individual, user-defined confidence degrees, taking up on our Remark 4.1. We also would like to directly learn a cost function that takes into consideration validity and efficiency [47] for a multi-target regression problem, possibly by using the hyper-rectangle volume as a parameter to define ϵ_t values that give us the smallest volume for the same validity.

Finally, while we mostly focused on multi-variate regression in the present paper, it would be interesting to try to extend the current approach to other multi-task settings, such as multi-label problems. A possibility could be to make such problems continuous, as proposed for instance by Liu [31].

6. Acknowledgments

This research was supported by the UTC foundation.

References

[1] G. Shafer, V. Vovk, A tutorial on conformal prediction, Journal of Machine Learning Research 9 (Mar) (2008) 371–421.

[2] H. Papadopoulos, K. Proedrou, V. Vovk, A. Gammerman, Inductive confidence machines for regression, in: European Conference on Machine Learning, Springer, 2002, pp. 345–356.

[3] R. Caruana, A dozen tricks with multitask learning, in: Neural networks: tricks of the trade, Springer, 1998, pp. 165–191.

[4] H. Wang, X. Liu, I. Nouretdinov, Z. Luo, A comparison of three implementations of multi-label conformal prediction, in: International Symposium on Statistical Learning and Data Sciences, Springer, 2015, pp. 241–250.

[5] R. Wang, S. Kwong, X. Wang, Y. Jia, Active k-labelsets ensemble for multi-label classification, Pattern Recognition (2020) 107583.

[6] K. A. Nguyen, Z. Luo, Reliable indoor location prediction using conformal prediction, Annals of Mathematics and Artificial Intelligence 74 (1) (2015) 133–153.

[7] M. Vrba, D. Heřt, M. Saska, Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system, IEEE Robotics and Automation Letters 4 (4) (2019) 3402–3409.

[8] A. Kuleshov, A. Bernstein, E. Burnaev, Conformal prediction in manifold learning, in: Conformal and Probabilistic Prediction and Applications, 2018, pp. 234–253.

[9] J. Neeven, E. Smirnov, Conformal stacked weather forecasting, in: Conformal and Probabilistic Prediction and Applications, 2018, pp. 220–233.

[10] H. Papadopoulos, V. Vovk, A. Gammerman, Regression conformal prediction with nearest neighbours, Journal of Artificial Intelligence Research 40 (2011) 815–840.

[11] S. Messoudi, S. Destercke, S. Rousseau, Conformal multi-target regression using neural networks, in: Conformal and Probabilistic Prediction and Applications, PMLR, 2020, pp. 65–83.

[12] R. B. Nelsen, An introduction to copulas, volume 139 of, Lecture Notes in Statistics.

[13] A. Gammerman, V. Vovk, V. Vapnik, Learning by transduction, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (1998) 148155.

[14] H. Papadopoulos, H. Haralambous, Reliable prediction intervals with regression neural networks, Neural Networks 24 (8) (2011) 842–851.

[15] U. Johansson, H. Linusson, T. Löfström, H. Boström, Interpretable regression trees using conformal prediction, Expert systems with applications 97 (2018) 394–404.

[16] M. Sánchez-Fernández, M. de Prado-Cumplido, J. Arenas-García, F. Pérez-Cruz, Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems, IEEE transactions on signal processing 52 (8) (2004) 2298–2307.

[17] G. De'Ath, Multivariate regression trees: a new technique for modeling species–environment relationships, Ecology 83 (4) (2002) 1105–1117.

[18] L. Baldassarre, L. Rosasco, A. Barla, A. Verri, Multi-output learning via spectral filtering, Machine learning 87 (3) (2012) 259–301.

[19] T. Aho, B. Ženko, S. Džeroski, Rule ensembles for multi-target regression, in: 2009 Ninth IEEE International Conference on Data Mining, IEEE, 2009, pp. 21–30.

[20] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: treating targets as inputs, Machine Learning 104 (1) (2016) 55–98.

[21] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint arXiv:1706.05098.

[22] R. Caruana, Multitask learning, Machine learning 28 (1) (1997) 41–75.

[23] M. Sklar, Fonctions de repartition an dimensions et leurs marges, Publ. inst. statist. univ. Paris 8 (1959) 229–231.

[24] M. Fréchet, Sur les tableaux de corrélation dont les marges sont données, Ann. Univ. Lyon, 3ˆ e serie, Sciences, Sect. A 14 (1951) 53–77.

[25] W. Höffding, Masstabinvariante korrelationstheorie, Schriften des Mathematischen Instituts und Instituts fur Angewandte Mathematik der Universitat Berlin 5 (1940) 181–233.

[26] W. Höffding, Masstabinvariante korrelationsmasse für diskontinuierliche verteilungen, Archiv für mathematische Wirtschafts-und Sozialforschung 7 (1941) 49–70.

[27] W. Höffding, Scaleinvariant correlation theory, in: The collected works of Wassily Höffding, Springer, 1994, pp. 57–107.

[28] P. Embrechts, A. McNeil, D. Straumann, Correlation and dependence in risk management: properties and pitfalls, Risk management: value at risk and beyond 1 (2002) 176–223.

[29] A.-C. Favre, S. El Adlouni, L. Perreault, N. Thiémonge, B. Bobée, Multivariate hydrological frequency analysis using copulas, Water resources research 40 (1).

[30] A. K. Nikoloulopoulos, D. Karlis, Multivariate logit copula model with an application to dental data, Statistics in Medicine 27 (30) (2008) 6393–6406.

[31] W. Liu, Copula multi-label learning, in: Advances in Neural Information Processing Systems, 2019, pp. 6337–6346.

[32] A. J. McNeil, R. Frey, P. Embrechts, Quantitative risk management: concepts, techniques and tools-revised edition, Princeton university press, 2015.

[33] T. Schmidt, Coping with copulas, Copulas-From theory to application in finance (2007) 3–34.

[34] E. J. Gumbel, Distributions des valeurs extremes en plusiers dimensions, Publ. Inst. Statist. Univ. Paris 9 (1960) 171–173.

[35] C. Genest, L.-P. Rivest, Statistical inference procedures for bivariate archimedean copulas, Journal of the American statistical Association 88 (423) (1993) 1034–1043.

[36] M. J. Frank, On the simultaneous associativity off (x, y) andx+y- f (x, y), Aequationes mathematicae 19 (1) (1979) 194–226.

[37] M. Hofert, I. Kojadinovic, M. Mächler, J. Yan, Elements of copula modeling with R, Springer, 2019.

[38] L. Ruschendorf, Asymptotic distributions of multivariate rank order statistics, The Annals of Statistics (1976) 912–923.

[39] F. H. Ruymgaart, Asymptotic theory of rank tests for independence, MC Tracts.

[40] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: Advances in neural information processing systems, 2017, pp. 971–980.

[41] F. Zhou, Q. Claire, R. D. King, Predicting the geographical origin of music, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 1115–1120.

[42] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, J. Huerta, Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems, in: 2014 international conference on indoor positioning and indoor navigation (IPIN), IEEE, 2014, pp. 261–270.

[43] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: A java library for multi-label learning, Journal of Machine Learning Research

12 (71) (2011) 2411–2414.
URL http://jmlr.org/papers/v12/tsoumakas11a.html

[44] C. Nugteren, V. Codreanu, Cltune: A generic auto-tuner for opencl kernels, in: 2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, IEEE, 2015, pp. 195–202.

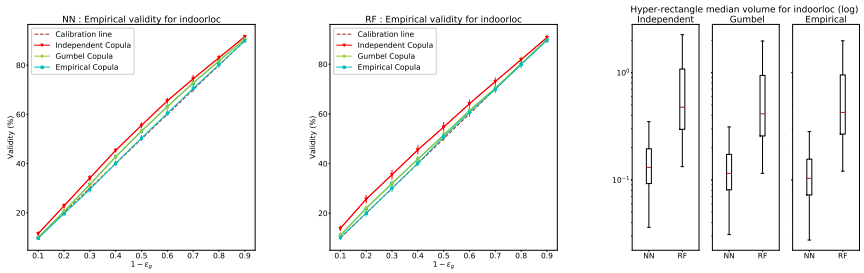
[45] V. Vovk, J. Shen, V. Manokhin, M.-g. Xie, Nonparametric predictive distributions based on conformal prediction, in: Conformal and Probabilistic Prediction and Applications, PMLR, 2017, pp. 82–102.

[46] H. Joe, D. Kurowicka, Dependence modeling: vine copula handbook, World Scientific, 2011.

[47] N. Colombo, V. Vovk, Training conformal predictors, in: Conformal and Probabilistic Prediction and Applications, PMLR, 2020, pp. 55–64.

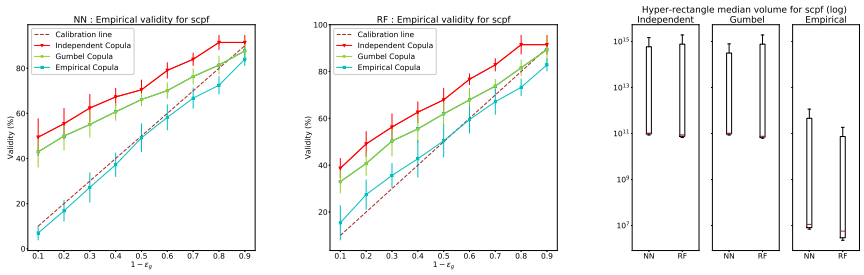
Appendix A. Validity and efficiency figures

This appendix contains the figures for empirical validity and hyper-rectangle median volume for all remaining data sets.



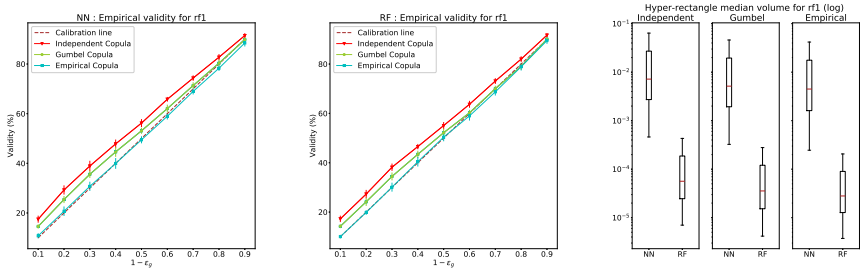
(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.5: Results for indoor loc.



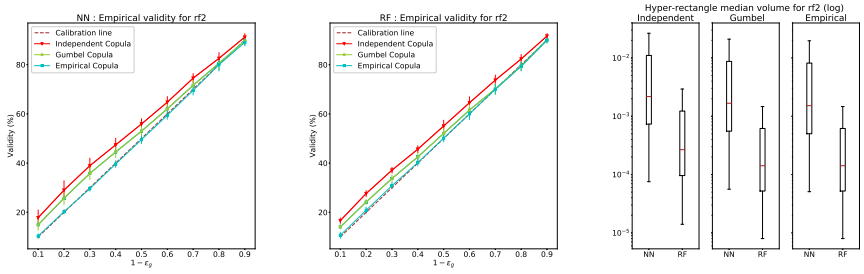
(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.6: Results for scpf.



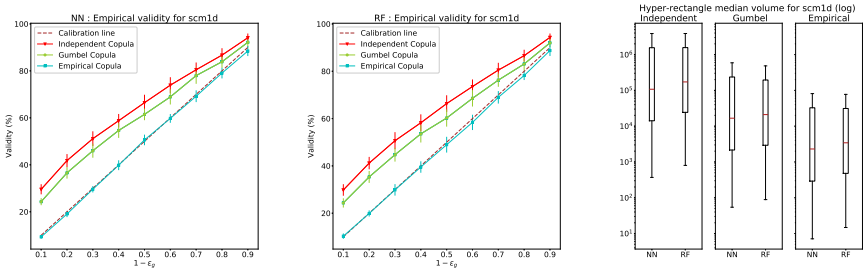
(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.7: Results for rf1.



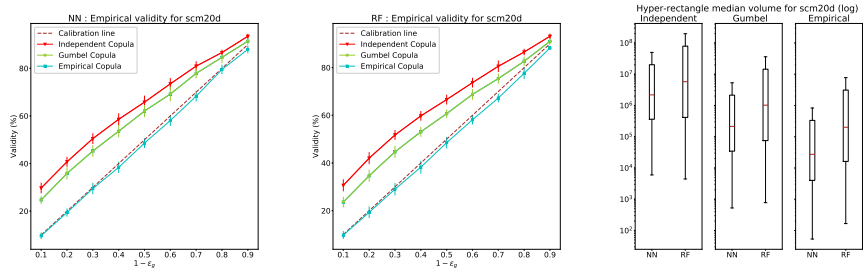
(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.8: Results for rf2.



(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.9: Results for scm1d.



(a) Validity (NN) (b) Validity (RF) (c) Efficiency (both)

Figure A.10: Results for scm20d.