Learning Calibrated Belief Functions from Conformal Predictions

Vitor Martin Bordini VITOR.MARTIN-BORDINI@HDS.UTC.FR
Sébastien Destercke SEBASTIEN.DESTERCKE@HDS.UTC.FR
Benjamin Quost BENJAMIN.QUOST@HDS.UTC.FR
Heudiasyc laboratory, UMR CNRS 7253, Université de Technologie de Compiègne, France

Abstract

We consider the problem of supervised classification. We focus on the problem of calibrating the classifier's outputs. We show that the p-values provided by Inductive Conformal Prediction (ICP) can be interpreted as a possibility distribution over the set of classes. This allows us to use ICP to compute a predictive belief function which is calibrated by construction. We also propose a learning method which provides p-values in a simpler and faster way, by making use of a multi-output regression model. Results obtained on the Cifar10 and Digits data sets show that our approach is comparable to standard ICP in terms of accuracy and calibration, while offering a reduced complexity and avoiding the use of a calibration set.

Keywords: conformal prediction, Dempster-Shafer, belief functions, calibration

1. Introduction

Nowadays, although many learning models perform well (i.e. they are able to well classify test instances into classes), their outputs are often ill-calibrated: the posterior probability distributions over the classes that they produce can be crude estimates of the actual ones. This is true irrespectively of whether those probabilities are directly produced by the classifier (as in the naive Bayes classifier) or are transformations of real-valued scores (as when using a softmax after having learned a deep learning model).

Inductive Conformal Prediction (ICP) [20, 18, 21], which received an increased interest in the past years, solves this problem by calibrating the model outputs, which results in sets of plausible classes rather than a single class. Usually, ICP is used within a probabilistic framework; however, conformal prediction and connected frameworks such as Venn predictors [15] are strongly linked to imprecise probabilistic models. This paper further explores and exploits the link between possibility theory and conformal prediction, something that was already pointed out in other setting such as self-supervised learning [16] or inferential models [3].

In particular, we want to tackle some of the drawbacks of conformal prediction by leveraging those links with possibility theory and belief functions. First, as other post-hoc calibrators [22], the approach requires a specific data set known as the "calibration set" in order to calibrate the classifier's outputs towards the desired posterior probabilities. Getting rid of this calibration set when making predictions may allow for not keeping it in store (thus saving memory space) or avoiding communicating it in further applications (thus increasing data privacy). Second, to classify a new instance, a set of conformity (or non-conformity) scores need to be computed via a specific model, which can be computationally expensive, and which makes ICP slower than a standard prediction method. While computing these scores is not computationally prohibitive (since it basically requires one sorting of p-values per possible class), it may still impede real-time applications such as in autonomous vehicles, high-speed processing lines, and all sorts of online detection or recommendation systems.

While our work can be totally put under the hood of possibility theory, we will consider the more general model of belief functions in this paper. One of the main reasons is that calibration issues have been much more investigated within the belief functions setting (see for example [5, 17, 29]) than in other uncertainty frameworks.

Our contributions are as follows: we provide a simple proof that so-called p-values produced by ICP can be directly interpreted as possibility degrees, allowing one to interpret ICP outputs within possibility and evidence theories[1]. We also investigate whether those p-values can be learned and reproduced in a reliable way, therefore solving the two issues mentioned above: the need for a calibration set, and preserving prediction efficiency. This means that we will only need the calibration set once, and in an off-line setting, in order to learn our p-value predictor. Provided that this latter is sufficiently accurate, combining those two elements then gives an easy practical means to obtain calibrated belief functions in the form of possibility distributions.

The rest of the paper is organized as follows. Conformal prediction is briefly presented in Section 2.2, and Section 2.3 provides a reminder on evidence theory and its relation to possibility theory. Section 3 formalizes our contributions.

[1] Since p-values are not necessarily normalised, they are harder to interpret directly as credal sets.

© 2023 V. Martin Bordini, S. Destercke & B. Quost.

Section 4 presents some experiments, and a conclusion is drawn in Section 5.

2. Preliminaries

This section provides the necessary preliminaries to follow the rest of the paper, and establishes notations.

2.1. Problem Setting

In this paper, we consider the classical problem of supervised classification: for each instance, a label must be specified among all possible classes. More formally, let $\Omega = \{w_1, \dots, w_K\}$ be the set of all possible classes (output space), and $\mathcal{X}$ the data (input) space. Let $Z = \{z_i = (x_i, y_i)\}$, $i = 1, \dots, \lambda$ be a labeled data set, with $x_i \in \mathcal{X}$ a data example and $y_i \in \Omega$ its associated label.

We assume that the data in Z are exchangeable (an assumption slightly weaker than iid), i.e, changing the order of the data set does not change the output of the model or our inferences. Among other things, this implies that the data Z are issued from the same distribution over $\mathcal{X} \times \Omega$, meaning that to each instance x can be associated a posterior probability distribution $\rho(w_k|x) \in \mathcal{P}(\Omega)$ giving the probability of any class $w_k \in \Omega$ given an input x.

The basic purpose in supervised classification is then to build a function $h : \mathcal{X} \longrightarrow \Omega$ that predicts well the class of new instances. However, such a function will not contain any uncertainty quantification of the output. A means to equip such a function with uncertainty quantification is to allow for set-valued predictions, that is to build a function $h : \mathcal{X} \longrightarrow 2^\Omega$ and to require that it satisfies some coverage properties, for instance that the predictions $h(X)$ satisfy

$$P(Y \in h(X)) > \alpha \tag{1}$$

for some specified values α. Such a constraint corresponds to a marginal coverage guarantee.

2.2. Inductive Conformal Prediction

Definition 1 *A model is said to be calibrated when its probability estimation is equal to the real one, i.e,*

$$P(w_i = y_i|h(x_i) = \alpha) = \alpha \tag{2}$$

Inductive conformal prediction (ICP) has been proposed to calibrate (probabilistic) classifier outputs. In order to apply ICP, the initial data set is split into a (proper) training set $\mathcal{D}_{tr}$, a calibration set $\mathcal{D}_{cal}$ and a test set $\mathcal{D}_{te}$, of respective sizes n, q and l ($n + q + l = \lambda$). ICP then outputs *conformal sets*, defined as follows.

Definition 2 *A conformal classifier evaluating a sample x_i predicts a conformal set $\Gamma^\delta \subseteq \Omega$, i.e. a set of possible labels such that:*

$$P(y_i \in \Gamma^\delta) \geq 1 - \delta, \tag{3}$$

where y_i is the true label and δ is a specified confidence (or significance) level.

It is easy to see that conformal sets satisfy Equation (1), and are therefore calibrated. This comes at a cost, since to determine these sets, ICP requires a separate calibration set $\mathcal{D}_{cal}$. Calibration instances are used to calculate *non-conformity scores* $\beta_i^{w_k}$ for all $w_k \in \Omega$. Such scores can be computed in various ways (see [20] for different proposals); in this paper, we use

$$\beta_i^{w_k} = \frac{\max_{j \neq k} o_i^{w_j}}{o_i^{w_k} + \epsilon}, \tag{4}$$

where $o_i = [o_i^{w_1}, \dots, o_i^{w_K}]$ is the raw (uncalibrated) vector predicted by the classifier for input x_i, and the positive $\epsilon \approx 0$ is meant to avoid division by zero.

The score of any calibration instance is computed for its actual class only. For any test instance, the scores are computed for all classes, so that they can be compared to those of the calibration data: thus, we can estimate to which extent the instance is typical of (or conformal to) the various classes. This can formally be done by computing *p-values* $p_i^{w_k}$ for any sample x_i outside of the calibration set:

$$p_i^{w_k} = \frac{\text{card}\left\{x_t \in \mathcal{D}_{cal} : \beta_t^{y_t} \geq \beta_i^{w_k}\right\}}{q+1}. \tag{5}$$

P-values have the following property [20]:

$$P(p^{w_k} \leq \delta) \leq \delta, \tag{6}$$

with $\delta \in (0, 1)$ being a significance level. Note that Equation (6) is exactly equivalent to the notion of *valid belief function* mentioned in [8] if we interpret p^{w_k} as plausibility degrees over the singletons. Therefore, p-values can be used as a statistic to decide whether a class label should be added to the conformal set: we define our set-valued prediction as

$$\Gamma^\delta(x_i) = \left\{w_j : p_i^{w_j} > \delta\right\}, \tag{7}$$

which satisfies the coverage constraint. The whole ICP process is summarised by Algorithm 1.

Conformal prediction sets can be made larger or smaller by varying δ: when applied to the whole set of test instances $\mathcal{D}_{te}$, we can estimate the resulting accuracy as

$$\nu_\delta = \frac{1}{n_{te}} \sum_{x_i \in \mathcal{D}_{te}} \mathbb{1}_{y_i \in \Gamma^\delta(x_i)}, \tag{8}$$

Algorithm 1 ICP algorithm

```
Input: Classifier $h$, calibration set $\mathcal{D}_{cal}$, test set $\mathcal{D}_{te}$
Output: Vector $p$ of p-values
for $x_i \in \mathcal{D}_{cal}$ do                              ▹ scores of calibration data
    $o_i \leftarrow h(x_i)$; $\beta_t^{y_t} \leftarrow \frac{\max_{j;j \neq t} o_t^{w_j}}{o_t^{y_t} + \epsilon}$;
end
for $x_i \in \mathcal{D}_{te}$ do                               ▹ scores of test data
    $o_i \leftarrow h(x_i)$;
    for $k \leftarrow 1$ to $K$ do
        $\beta_i^{w_k} \leftarrow \frac{\max_{j;j \neq k} o_i^{w_j}}{o_i^{w_k} + \epsilon}$;
        $p_i^{w_k} \leftarrow \frac{|t=n+1,\dots,n+q;\beta_t^{y_t} \geq \beta_i^{w_k}|}{q+1}$;
    end
end
```

with $\mathbb{1}_A$ the indicator function of event A, i.e., the function that equals 1 if A is true, zero else. If a model is well calibrated, its accuracy ν_δ is expected to be δ, for all $\delta \in [0;1]$. The graph representing ν_δ as a function of δ is called a *validation curve*.

Example 1 *Consider that, for a given classifier h predicting four classes $\Omega = \{w_1, w_2, w_3, w_4\}$, the following non-conformity scores were computed over the calibration set containing 7 data points:*

$$[0.8, 2, 3.15, 1.5, 1, 6.4, 5.8]$$

For a given instance x_i, the same classifier provides $o_i = [0.6, 0.3, 0.2, 0.1]$ as the class probabilities, which is then turned into the following non-conformity scores using Equation (4):

$$\beta_i = [0.5, 2, 3, 6].$$

Applying Equation (5) gives the following p-values:

$$\begin{aligned} p_i &= [7/7{+}1, 4/7{+}1, 3/7{+}1, 1/7{+}1] \\ &= [0.875, 0.5, 0.375, 0.125]. \end{aligned}$$

Assuming we specify $\delta = 0.2$, that is we require a coverage of 80%, we would then obtain as prediction

$$\Gamma^\delta(x_i) = \{w_1, w_2, w_3\}.$$

By increasing δ, Γ^δ tends to have more elements, until it eventually becomes Ω, i.e, all possible classes. This is illustrated by Figure 1.

While ICP is a versatile and efficient tool to provide cautious predictions, its reliance on a calibration data set can be seen as a limitation: one must keep the calibration data at disposal, and producing the conformal prediction scores still requires a number of computations that may prevent its application to real-time problems. In the following, we see ICP as an intermediate way to obtain a predictive model outputting calibrated belief functions. Before doing so, we will recall the basics about possibility distributions and belief functions.

2.3. Belief Functions and Possibility Theory

2.3.1. Belief Functions

The theory of belief functions, a.k.a. Dempster-Shafer (DS) theory or evidence theory, is a useful tool to represent and manage the partial knowledge of an unknown variable (e.g., a class variable $Y \in \Omega$ as above) [7].

The basic representation is that of mass function.

Definition 3 *A mass function (MF) is a mapping $m : 2^\Omega \rightarrow [0, 1]$ such that*

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{9}$$

Each mass $m(A)$ can be interpreted as a piece of evidence that $Y \in A$. If $m(A) > 0$, then A is said to be a focal set of m. A MF is normal if $\emptyset$ is not a focal set. It is consonant if all the focal sets are nested, i.e. either $A_i \subset A_j$ or $A_j \subset A_i$ for $A_i \neq A_j$.

We also introduce the belief and plausibility functions.

Definition 4 *A MF is in one-to-one correspondence with its associated belief and plausibility functions:*

$$Bel(B) = \sum_{B \subseteq A} m(A), \tag{10a}$$

$$Pl(B) = \sum_{A \cap B \neq \emptyset} m(A) = Bel(\Omega) - Bel(\neg B). \tag{10b}$$

When a MF is normal, $Bel(\Omega) = 1$ and the belief and plausibility degrees can be interpreted as lower and upper bounds on the probability of any $A \subseteq \Omega$:

$$Bel(A) \leq P(A) \leq Pl(A).$$

Because of their duality (10b), we need only use one of them to define the associated credal set of probability distributions:

$$\mathcal{P}(Pl) = \{P | \forall p(A) \leq Pl(A)\}. \tag{11}$$

Last, the *contour function* $pl : \Omega \rightarrow [0;1]$ can be defined from Pl by $pl(w) = Pl(\{w\})$, for all $w \in \Omega$.

2.3.2. Possibility Theory

Another framework for representing and managing uncertainty, possibility theory [11, 8] has strong connections with belief functions. Its basic representation tool is that of possibility distribution.

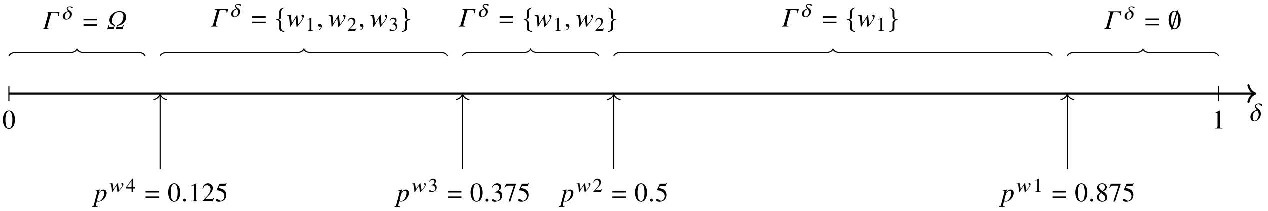

$\Gamma^\delta = \Omega$
$\Gamma^\delta = \{w_1, w_2, w_3\}$
$\Gamma^\delta = \{w_1, w_2\}$
$\Gamma^\delta = \{w_1\}$
$\Gamma^\delta = \emptyset$
0
1
δ
$p^{w4} = 0.125$
$p^{w3} = 0.375$
$p^{w2} = 0.5$
$p^{w1} = 0.875$

Figure 1: Illustration of predicted conformal sets

Definition 5 *A possibility distribution π is a mapping $\pi : \Omega \to [0,1]$.*

From a possibility distribution can again be defined two dual measures, the possibility and necessity measures.

Definition 6 *The possibility and necessity measures associated with a possibility distribution π are mappings $\Pi : 2^\Omega \to [0,1]$ and $N : 2^\Omega \to [0,1]$, such that, for any $B \subseteq \Omega$,*

$$\Pi(B) = \max_{x \in B} \pi(x), \quad N(B) = \Pi(\Omega) - \Pi(\neg B).$$

The possibility measure is maxitive, in the sense that $\Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\}$. When the possibility distribution is normalised, that is when $\max_{x \in \Omega} \pi(x) = 1$, the possibility measure furthermore satisfies

$$\Pi(\emptyset) = 0, \ \Pi(\Omega) = 1,$$

and in this case the necessity and possibility degrees can also be seen as lower and upper bounds of a probability: $N(A) \leq P(A) \leq \Pi(A)$. More generally, they correspond to the belief and plausibility functions of a consonant mass function. Another notion issued from possibility theory that we will use is the one of α-cut: the alpha-cut of a distribution π is the subset

$$\pi_\alpha = \{w \in \Omega : \pi(w) \geq \alpha\}.$$

Let us also recall that if π is normalised, we have that

$$N(\pi_\alpha) = 1 - \alpha.$$

3. Calibrated Belief Functions through ICP

In this section, we will show that we can generally interpret ICP results as consonant belief functions or possibility distributions, and will investigate whether we can learn these latter. Note that the relation we will show was already known in the framework of inferential models [3]. Our derivation is nevertheless simpler.

3.1. P-values as Possibility Degrees

Let x be an instance and p^{w_j}, $j = 1, \ldots, K$ be the p-values obtained from its nonconformity scores, and assume that those p-values are normalised. We propose to derive a possibility distribution from these latter.

Proposition 1 *The p-values obtained by ICP can be interpreted as a possibility distribution π_x defined by $\pi_x(w_j) = p^{w_j}$ for all $w_j \in \Omega$, the α-cuts of which are the conformal sets obtained from the p^{w_j}.*

Proof First, we obviously have $p^{w_j} \in [0;1]$ for all $j = 1, \ldots, K$, with p^{w_j} being as large as w_j is a plausible label. Thus, the set of degrees $\pi_x(w_j) = p^{w_j}$ defines a legitimate possibility distribution. Note that this latter is in general unnormalized.

In addition, we recall that according to (6),

$$P(p^{w_j} \leq \delta) \leq \delta \quad \Leftrightarrow \quad P(p^{w_j} > \delta) \geq 1 - \delta;$$

this means that the set $\Gamma^\delta(x)$ with confidence level $1 - \delta$ defined by Eq. (7) can be rewritten as

$$\Gamma^\delta(x) = \left\{ w_j : \pi_x(w_j) \geq \delta \right\}, \tag{12}$$

which is the definition of the (δ)-cut of the possibility distribution π_x. ■

In general, p-values will not be normalised though, and the possibility distribution derived from them will not correspond to a probability set.

3.2. Normalised ICP

ICP is first applied to obtain the p-values for each class. These p-values can be interpreted as a possibility distribution π_x as described in Proposition 1; however they are generally not normalised, in the sense that $\max_{w \in \Omega} \pi_x(w) < 1$. In practice, one may wish to handle normalised distributions for various reasons:

- First, one may wish to output a non-empty conformal set $\Gamma^\delta(x)$ whichever δ in Equation (12)—unless $\Gamma^\delta(x) = \emptyset$ can be given a meaningful interpretation;

- Second, being able to interpret the prediction as a probability set may be necessary or useful, for instance to use various decision rules [5, 26] or to use learning methods using credal labels [16].

In our case, the normalisation strategy should preserve the conformity property given by Definition 2. This means in particular that if $\Gamma^{\delta}_{\pi}(x)$ are the sets obtained from π_x, then the sets $\Gamma^{\delta}_{\pi'}(x)$ given by the normalised π'_x should satisfy $\Gamma^{\delta}_{\pi}(x) \subseteq \Gamma^{\delta}_{\pi'}(x)$ for all $\delta \in [0,1]$. This ensures that if our initial predictions were valid in the sense of Definition 2, the new ones will be too, albeit in a more conservative way. Note that having $\Gamma^{\delta}_{\pi}(x) \subseteq \Gamma^{\delta}_{\pi'}(x)$ for all $\delta \in [0,1]$ is equivalent to require $\pi \leq \pi'$: hence, any normalisation strategy giving such a π' can be considered.

However, as we said, such π' will provide more cautious and conservative predictions. It is therefore natural to limit as much as possible this increase in conservativeness, as the conformal procedure already guarantees predictions to be well-calibrated. Therefore, a second natural desiderata in our setting is that the normalised π' should be as close as possible to the original π.

In order to satisfy these two conditions, we propose to normalise the distribution by changing only the maximum p-value for one, i.e

$$\pi^*_x(w^*) = 1, \tag{13a}$$
$$\pi^*_x(w) = \pi_x(w), \quad \text{for all } w \neq w^*; \tag{13b}$$

with $w^* = \arg\max(\pi(w))$. The normalization in Eq. (13) amounts to assume that the most plausible class is indeed completely plausible, i.e. $\Pi(\Omega) = 1$.

It should be noted that such a normalisation is uncommon in both evidence theory and possibility theory. It can be compared to two common other normalisation rules in evidence theory: Dempster's normalisation, which results in

$$\pi_d(w) = \frac{\pi(w)}{\max_{w\in\Omega} \pi(w)},$$

and Yager's normalisation, which amounts to compute

$$\pi_y(w) = \pi(w) + (1 - \max_{w\in\Omega} \pi(w)).$$

The former has for instance been used in previous works connected to conformal prediction [16, 3], however it should be noted that for a given π we have $\pi^* \leq \pi_d \leq \pi_y$. In our setting, we will therefore normalise according to Equation (13) so that π_x is altered in a minimal way.

Finally, the normalized possibility distribution is interpreted as a plausibility distribution, and the corresponding belief function can be retrieved:

$$Bel(B) = 1 - \Pi^*(\neg B) = 1 - \max_{x\in\neg B} \pi^*(x), \forall B \subseteq \Omega. \tag{14}$$

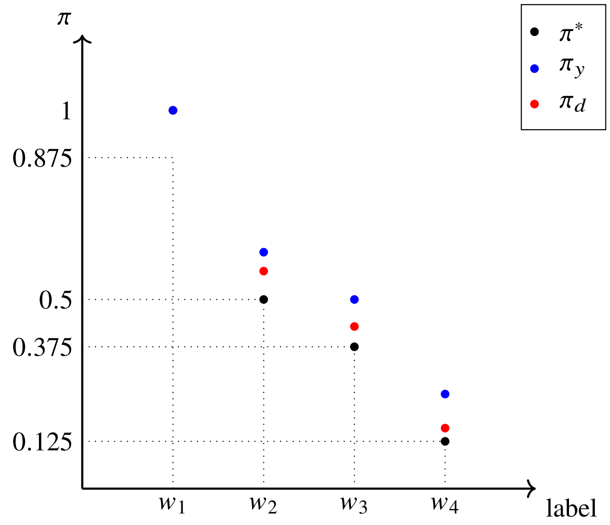


Figure 2: Illustration of different normalisation procedures.

Note that the associated mass function is consonant [10]. A direct consequence is that $Bel(B) = 0$ for all $B \subseteq \Omega$ such that $w^* \notin B$, since $1 - \max_{x\in\neg B} \pi^*(x) = 1 - 1 = 0$.

Example 2 *Following Example 1, the unnormalized possibility distribution derived from the p-values at hand is*

$$\pi(w_1) = 0.875, \quad \pi(w_2) = 0.5,$$
$$\pi(w_3) = 0.375, \quad \pi(w_4) = 0.125,$$

with π^* being obtained by setting $\pi^*(w_1) = 1$ and $\pi^*(w_j) = \pi(w_j)$ for all $j \neq 1$. The associated belief function is

$$\begin{aligned}
Bel(\{w_1\}) &= 1 - 0.5 = 0.5,\\
Bel(\{w_1, w_2\}) &= 1 - 0.375 = 0.625,\\
Bel(\{w_1, w_3\}) &= 1 - 0.5 = 0.5,\\
Bel(\{w_1, w_4\}) &= 1 - 0.5 = 0.5,\\
Bel(\{w_1, w_2, w_3\}) &= 1 - 0.125 = 0.875,\\
Bel(\{w_1, w_2, w_4\}) &= 1 - 0.375 = 0.625,\\
Bel(\{w_1, w_3, w_4\}) &= 1 - 0.5 = 0.5,\\
Bel(\Omega) &= 1.
\end{aligned}$$

As expected, whenever $w_1 \notin B$, we have $Bel(B) = 0$.

Figure 2 represents the three possible normalisations: ours, Yager's and Dempster's. It clearly shows how these two latter alter all p-values, contrary to the former.

Normalizing the possibility distribution in this way means that instead of returning the empty set $\emptyset$ for higher values of δ, one will return the most likely class. This may be perceived as a reasonable behaviour in case the true class is assumed to be in Ω. In any case, this can only improve the coverage, as we will see in the experiments.

3.3. Predicting p-Values

We have now seen that computing p-values directly gives a belief function (or a possibility distribution) representing our partial knowledge of the actual class label of an instance. Therefore, being able to regress accurately such p-values given any new test instance, e.g. using a multi-output model, would provide us with a credal classifier with built-in calibration guarantees. This would also allow us to avoid having to repeatedly use the calibration set, possibly saving some precious time at the prediction step.

Since p-values are not available in classical data sets, we propose to first train a probabilistic model $h : \mathcal{X} \longrightarrow \mathcal{P}(\Omega)$ so as to retrieve scores for any processed instance. Once this model has been trained, ICP can be used to transform the scores obtained for the training instances into p-values p_{tr}. Using the newly labeled training data $(\mathcal{D}_{tr}, p_{tr})$, we can consequently train a regression function (or regressor) $\tau : \mathcal{X} \longrightarrow [0, 1]^K$ to directly provide p-values from any instance $x \in \mathcal{X}$, using the procedure summarized in Algorithm 2.

Computing p-values using this regressor is obviously faster than computing scores before applying ICP. If we suppose that the regressor and the probabilistic classifier have the same complexity, we nevertheless decrease the overall computational cost, since applying the procedure described in Section 3.2 onto the estimated p-values $\tau(x)$ provides us with a calibrated belief function for any test instance x, without having to compare its nonconformity scores to those of the calibration data.

4. Experiments

We validated the procedure in Algorithm 2 through two experiments realized on real data sets, briefly presented in Table 1. They exhibit relatively low-dimension inputs. The “input shape” refers to the Height × Width × Number of the channels for the Digits, Cifar10, Cifar100, Artists and SVHN data sets since they are images. For the rest of the data sets, it is the size of a feature vector.

For the first four data sets, we used a logistic regression model as classifier h and a random forest [2] as regression model. For the others, we used an EfficientNet.v2 neural network [24] coupled with a softmax layer for feature extraction and classification, the regressor being implemented using three dense linear layers. We used various classifiers and regressors in order to test the applicability of our approach in different settings.

The following parameters were used for training in the second experiment: we used an Adam optimizer, with batch size 128, initial learning rate 0.001 and momentum 0.9. An exponential learning rate schedule was created. The calibration data set for all experiments was set as 20% of the training data set.

Algorithm 2 Training algorithm

```
Input: Training set $\mathcal{D}_{tr}$, calibration set $\mathcal{D}_{cal}$, test set $\mathcal{D}_{te}$,
       number of epochs $n_e$, number of batches $n_b$
Output: Trained classifier $\widehat{h}$, trained regressor $\widehat{\tau}$
split $\mathcal{D}_{tr}$ into $n_b$ batches $\mathcal{D}_{tr}^b$ $(b = 1, \dots, n_b)$
$\widehat{a} \leftarrow 0$;                                        ▹ best accuracy
for $i \leftarrow 0$ to $n_e$ do
    for $j \leftarrow 1$ to $n_b$ do
        $L_c \leftarrow$ classification_loss$(h, \mathcal{D}_{tr}^b)$;
        $W \leftarrow W - \eta \frac{\partial L_c}{\partial W}$;                  ▹ update classifier
    end
    $a \leftarrow$ classification_accuracy$(h, \mathcal{D}_{te})$;
    if $a > \widehat{a}$ then
        $\widehat{h} \leftarrow h$;
        $\widehat{a} \leftarrow a$;
    end
end
$p_{tr} \leftarrow ICP(\widehat{h}, \mathcal{D}_{tr}, \mathcal{D}_{cal})$;        ▹ p-values of training data
$p_{te} \leftarrow ICP(\widehat{h}, \mathcal{D}_{te}, \mathcal{D}_{cal})$;            ▹ p-values of test data
$\widehat{\varepsilon} \leftarrow \infty$;
for $i \leftarrow 0$ to $n_e$ do
    for $j \leftarrow 1$ to $n_b$ do
        $L_r \leftarrow$ regression_loss$(\tau, \mathcal{D}_{tr}^b, p_{tr}^b)$;
        $W \leftarrow W - \eta \frac{\partial L_r}{\partial W}$;                   ▹ update regressor
    end
    $\varepsilon \leftarrow$ regression_error$(\tau, \mathcal{D}_{te}, p_{te})$;
    if $\widehat{\varepsilon} < \varepsilon$ then
        $\widehat{\tau} \leftarrow \tau$;
        $\widehat{\varepsilon} \leftarrow \varepsilon$;
    end
end
```

data set	# training instances	# test instances	Input Shape	# classes
Digits [9]	1437	360	(8,8,1)	10
Heart disease [9]	771	147	(9,1)	2
Titanic [13]	748	143	(7,1)	2
Symptom2Disease [1]	960	240	(384,1)	24
SVHN [19]	1437	360	(32,32,3)	10
Cifar10 [14]	50,000	10,000	(32,32,3)	10
Cifar100 [14]	50,000	10,000	(32,32,3)	100
Artists [12]	6700	1676	(512,512,3)	50

Table 1: data set description

4.1. P-Value Regression

For both experiments, we compared the p-value estimates provided by the trained regressor model τ and the actual p-values obtained via ICP. The Root Mean Square Residual

data set	Classifier	Regressor	Classifier accuracy	RMSR (10^{-3})	R2 coeff.
Digits	logistic regression	random forest	96	3.6	0.84
Heart			82	3.6	0.96
Titanic			80	0.9	0.99
Symptom2Disease			96	7	0.83
SVHN	EfficientNet.v2	feature extractor + 4 dense linear layers	94	0.03	0.99
Cifar10			85	0.34	0.98
Cifar100			62	9.80	0.17
Artists			89	0.80	0.78

Table 2: RMSR and R2 coefficients

(RMSR) and the R2 coefficient, displayed in Table 2, were computed for this purpose (our goal is to be as close as possible to $RMSR = 0$ and $R2 = 1$). They were computed by the following formulas:

$$RMSR = \sqrt{\frac{\sum_{i=1}^{N} ||\tau(x_i) - p_i||^2}{N}},$$

$$R2 = \frac{1}{K} \sum_{k=1}^{K} \left(1 - \frac{\sum_{i=1}^{N} \left(\tau_k(x_i) - p_i^{w_k}\right)^2}{\sum_{i=1}^{N} \left(p_i^{w_k} - \frac{1}{N} \sum_{n=1}^{N} p_i^{w_k}\right)^2} \right),$$

where $\tau_k(x_i)$ is the p-value estimated by the regressor for input x_i and kth class.

The results obtained with both metrics suggest that our method can effectively estimate the p-values (low RMSR and high R2). However, our approach seems to be less efficient as the number of classes increases, since the RMSR increases and the R2 decreases—for example, the results on the binary Titanic data ($R2 = 0.99$ and $RMSR = 0.9 * 10^{-3}$) and the 10-class Digits data ($R2 = 0.84$ and $RMSR = 3.6 * 10^{-3}$) are quite good, whereas we can see a decline in performance for the Cifar100 and Artists data (in particular according to the $R2$ metric). This is patent for Cifar100, for which the $R2$ result is disastrous.

4.2. Validation Curves

In order to empirically check that calibration is maintained by our learning algorithm, we display the validation curves in Figure 3. In each graph, five curves are provided for each data set: the ideal $y = x$ curve (purple), and the curves obtained with ICP (blue), our method (orange), ICP with normalization (green) and our method with normalization (red) as per Eq. (13). We can see that both ICP and our method appear to be well-calibrated before normalization. After normalization, the curve is of course higher, due to our choice which transforms the empty-set predictions into predicting the most plausible class.

This however points out a strange behaviour of conformal prediction, which may increase the prediction of empty sets in order to obtain the desired coverage guarantee, when this guarantee is low. In practice, this happens mostly when the coverage rate is smaller than the average accuracy obtained by the initial classifier, hence the threshold effect observed on all the curves. Instead, the normalized possibility distributions will always provide a minimal accuracy and mostly precise predictions, unless we specify a very high coverage rate, at which point imprecise predictions will be produced. Incidentally, this threshold effect is less severe in our case, as the change in coverage before reaching the classifier base accuracy is more gradual. Our results suggest that using conformal prediction with a guarantee level below the base accuracy of the classifier is actually not useful, as it would just transform singleton predictions into empty sets in order to lower the accuracy.

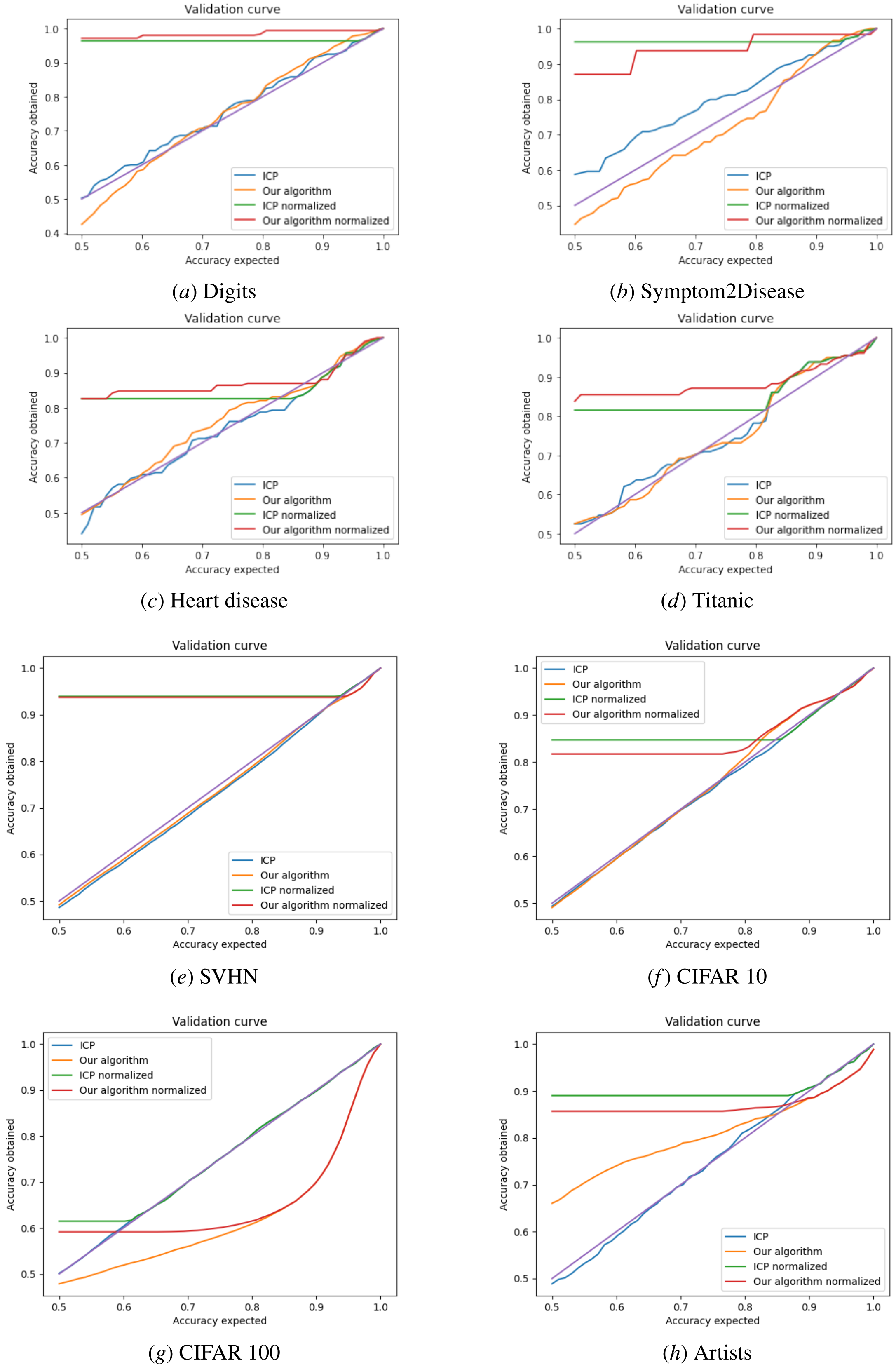
Remark also that our surrogate regression method does not perform well if the number of classes is high, as can be seen by the performances obtained on the Cifar100 and (to a lesser extent) Artist data. A possible explanation is that estimating the p-values then becomes difficult, i.e the features extracted cannot accurately represent the inputs.

5. Conclusion

In this work, we address the problem of learning a calibrated credal classifier able to make decisions with statistical guarantees. Inductive conformal prediction is one of the most popular techniques for this purpose; this approach nevertheless has drawbacks, among which its need for a large amount of data and its high computational cost. Our learning procedure aims at solving these issues by learning a surrogate regression model which directly estimates the p-values otherwise provided by ICP, from which calibrated belief functions can then be derived.

Nevertheless, our algorithm also has limitations. First, it currently works only for classification problems—the extension to other learning settings, such as regression, will be considered later. Second, it seems to be sensitive to the number of classes in the data. This could be solved by considering other models (or of course gathering more training data).

One of our future works is to use the evidential labels produced by our approach in different learning settings where taking into account label uncertainty could help improving model robustness. Learning from evidential labels is indeed an active research area [6, 4]; yet, how to produce or obtain such evidential labels is still an open question [25]. Our goal is to estimate calibrated evidential labels on unlabelled data, so as to integrate them in the learning process. More specifically, we aim at introducing evidential (or more generally credal) labels in self-training or co-learning strategies [23, 27]. Such strategies consist in repeating (i) classifying unlabeled instances and (ii) using these labeled data to further train the classifier. Taking into account classification uncertainty seems paramount to achieve a good classifica-



(*a*) Digits

(*b*) Symptom2Disease

(*c*) Heart disease

(*d*) Titanic

(*e*) SVHN

(*f*) CIFAR 10

(*g*) CIFAR 100

(*h*) Artists

Figure 3: Validation curves for all data sets.

tion accuracy, by avoiding undesired biases due to the lack of data or inappropriate model assumptions.

Finally, it would also be useful to compare the current approach with other evidential calibration methods, such as the one developed in [28] which extend classical scaling techniques.

References

[1] Nyar R Barman, Faizal Karim, and Krish Sharma. Symptom2disease. https://www.kaggle.com/datasets/niyarrbarman/symptom2disease. Accessed: 2023-04-20.

[2] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, oct 2001. ISSN 0885-6125. doi:10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[3] Leonardo Cella and Ryan Martin. Validity, consonant plausibility measures, and conformal prediction. *International Journal of Approximate Reasoning*, 141: 110–130, 2022.

[4] Thierry Denœux. Likelihood-based belief function: justification and some extensions to low-quality data. *International Journal of Approximate Reasoning*, 55 (7):1535–1547, 2014.

[5] Thierry Denœux. Decision-making with belief functions: a review. *International Journal of Ap-proximate Reasoning*, 109, 2019. doi:10.1016/j.ijar.2019.03.009ï. URL https://hal.archives-ouvertes.fr/hal-02471545.

[6] Thierry Denœux. Belief functions induced by random fuzzy sets: A general framework for representing uncertain and fuzzy evidence. *Fuzzy Sets and Systems*, 424:63–91, 2021.

[7] Thierry Denœux, Didier Dubois, and Henri Prade. Representations of uncertainty in AI: beyond probability and possibility. *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*, pages 119–150, 2020.

[8] Thierry Denœux and Shoumei Li. Frequency-calibrated belief functions: Review and new insights. *International Journal of Approximate Reasoning*, 92:232–254, 2018. ISSN 0888-613X. doi:https://doi.org/10.1016/j.ijar.2017.10.013. URL https://www.sciencedirect.com/science/article/pii/S0888613X17306448.

[9] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[10] Didier Dubois and Henri Prade. On several representations of an uncertain body of evidence. In M. M. Gupta and E. Sanchez, editors, *Fuzzy Information and Decision Processes*, pages 167–181, 1982.

[11] Didier Dubois and Henri Prade. *Possibility theory: an approach to computerized processing of uncertainty*. Springer Science & Business Media, 2012.

[12] Icaro. Best artworks of all time. https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time. Accessed: 2023-04-20.

[13] Will Cukierski Jessica Li. Titanic - machine learning from disaster, 2012. URL https://kaggle.com/competitions/titanic.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[15] Antonis Lambrou, Ilia Nouretdinov, and Harris Papadopoulos. Inductive venn prediction. *Annals of Mathematics and Artificial Intelligence*, 74:181–201, 2015.

[16] Julian Lienen, Caglar Demir, and Eyke Hüllermeier. Conformal credal self-supervised learning. *arXiv preprint arXiv:2205.15239*, 2022.

[17] Ryan Martin. False confidence, non-additive beliefs, and valid statistical inference. *International Journal of Approximate Reasoning*, 113:39–73, 2019.

[18] Soundouss Messoudi, Sylvain Rousseau, and Sébastien Destercke. Deep conformal prediction for robust models. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume 1237 CCIS, pages 528–540. Springer, 2020. ISBN 9783030501457. doi:10.1007/978-3-030-50146-4_39.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. 01 2011.

[20] Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. In Paula Fritzsche, editor, *Tools in Artificial Intelligence*, chapter 18. IntechOpen, Rijeka, 2008. doi:10.5772/6078. URL https://doi.org/10.5772/6078.

[21] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Machine Learning Research*, 2007. doi:10.48550/ARXIV.0706.3188. URL https://arxiv.org/abs/0706.3188.

[22] Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, Peter Flach, et al. Classifier calibration: How to assess and improve predicted class probabilities: a survey. *arXiv preprint arXiv:2112.10327*, 2021.

[23] Yann Soullard, Sébastien Destercke, and Indira Thouvenin. Co-training with credal models. In *Artificial Neural Networks in Pattern Recognition*, volume 9896, pages 92–104, 09 2016. ISBN 978-3-319-46181-6. doi:10.1007/978-3-319-46182-3_8.

[24] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. *International Conference on Machine Learning*, 2021. doi:10.48550/ARXIV.2104.00298. URL https://arxiv.org/abs/2104.00298.

[25] Constance Thierry, Arthur Hoarau, Arnaud Martin, Jean-Christophe Dubois, and Yolande Le Gall. Real bird dataset with imprecise and uncertain values. In *Belief Functions: Theory and Applications: 7th International Conference, BELIEF 2022, Paris, France, October 26–28, 2022, Proceedings*, pages 275–285. Springer, 2022.

[26] Matthias CM Troffaes. Decision making under uncertainty using imprecise probabilities. *International journal of approximate reasoning*, 45(1):17–29, 2007.

[27] Yingda Xia, Dong Yang, Zhiding Yu, Fengze Liu, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. *Medical Image Analysis*, 10 2020. ISSN 13618423. doi:10.1016/j.media.2020.101766.

[28] Philippe Xu, Franck Davoine, Hongbin Zha, and Thierry Denœux. Evidential calibration of binary SVM classifiers. *International Journal of Approximate Reasoning*, 72:55–70, 2016.

[29] Jianchun Zhang and Chuanhai Liu. Dempster-Shafer inference with weak beliefs. *Statistica Sinica*, pages 475–494, 2011.