Approximation to Object Conditional Validity with Inductive Conformal Predictors

Anthony Bellotti ANTHONY-GRAHAM.BELLOTTI@NOTTINGHAM.EDU.CN
School of Computer Science, University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo 315100, China

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

Conformal predictors are machine learning algorithms that output prediction sets that have a guarantee of marginal validity for finite samples with minimal distributional assumptions. This is a property that makes conformal predictors useful for machine learning tasks where we require reliable predictions. It would also be desirable to achieve conditional validity in the same setting, in the sense that validity of the prediction intervals remains true regardless of conditioning on any particular property of the object of the prediction. Unfortunately, it has been shown that such conditional validity is impossible to guarantee for non-trivial prediction problems for finite samples. In this article, instead of trying to achieve a strong conditional validity guarantee, an *approximation* to conditional validity is considered and measured empirically. A new algorithm is introduced to do this by iteratively adjusting a conformity measure to deviations from object conditional validity measured in the training data. Experimental results are provided for three data sets that demonstrate (1) in real world machine learning tasks, lack of conditional validity is a measurable problem and (2) that the proposed algorithm is effective at alleviating this problem.

Keywords: Conformal prediction, conditional validity, approximation.

1. Introduction

In some predictive analytic tasks it is useful to provide a prediction set of possible outcomes rather than a single point prediction. For example, a doctor diagnosing a patient's symptoms may find it more valuable if an automated decision support system suggested two or more possible causes instead of just one, if there is some ambiguity in diagnosis (Nouretdinov et al., 2014). Or, given the uncertainty in property sale prices, it may be better if an automated valuation model provides a range of possible predicted sale prices, rather than a single point estimate. Indeed, for automated valuation, the delivery of a property price range is the norm; see e.g. Bellotti (2017) or `www.zoopla.co.uk`.

The conformal predictor (CP) is a machine learning algorithm that is able to generate prediction sets with a guarantee that its prediction sets are correct, i.e. the true label [1] will be a member of the prediction set, at a preset confidence level (Vovk et al., 2005). We

1. We use the convention that each entity for which a prediction is to be made is called an *example*, with the *label* being the outcome we wish to predict and *object* the collection of features from which we may make a prediction, following Vovk (2014).

therefore say that the CP is a *valid* predictor. Other statistical methods based on frequentist and Bayesian approaches exist that promise the same guarantee, but only with some specific distributional assumptions. When those assumptions do not hold true, validity is no longer guaranteed. In contrast, CPs require only the assumption of exchangeability of data. Hence the CP is referred to as *distribution-free* (Lei and Wasserman, 2014). Central to the CP is the use of a conformity measure (CM) that, given a new example, provides a measure of how typical it is of the population. Often the CM is based on some underlying machine learning algorithm, although this is not essential. The CP has been applied successfully in many problem domains including medical and financial, anomaly detection and network traffic applications, amongst others; see e.g. Balasubramanian et al. (2014).

Since the *success rate*, i.e. the rate at which CP gives correct prediction sets, is essentially fixed by the validity result, the optimization task is to maximize the *efficiency* of the prediction sets which is some measure of the information given by the prediction set. There are multiple alternative measures of predictive efficiency, or conversely, predictive inefficiency (Vovk et al., 2016). For example, Colombo and Vovk (2020) optimize on the object fuzziness of the predictions. Another approach is to build CPs that minimize the mean width of prediction intervals for regression (Bellotti, 2020) or mean prediction set size for classification. This is also the approach used for Prediction Interval Neural Networks (Khosravi et al., 2011; Pearce et al., 2018). For clarification, Examples 1 and 2 give two cases of what is meant by predictive efficiency.

The CP guarantees *marginal validity*: that is, across the whole population of observations, the success rate of the CP will meet the confidence level preset by the user. However, even with marginal validity, there may be some sub-segments of the population for which success rate is greater or less than the confidence level. Here are two examples demonstrating why this may be a problem.

- *Medical example.* For diagnosing a disease, we could have a CP that gives 95% success rate guaranteed across all people. However, for a particular racial minority group, it gives only 85% success rate, whilst for the majority racial group it gives 96% success rate. This is clearly a problem for two reasons: (1) it makes clinical decisions less precise, especially for hospitals operating with a higher racial minority demographic, (2) it leads to medical decision support that is discriminatory.

- *Business example.* Suppose we have an automated valuation model (AVM) that gives good property price prediction intervals at 90% confidence levels across UK. However, for the particular county of Devon its success rate is only 75%. This could adversely affect the business of any real estate company or mortgage provider operating in Devon who use the AVM. Worst still, the weakness of the AVM in their location may not even be known and so they will be operating under the misconception that the AVM is guaranteed to be 90% success rate: it is in general, but not for them. A good example of this problem is given by Lim and Bellotti (2021) (Table 6) based on experiments on US real estate data.

For this reason, *conditional validity* has been identified as an important criterion, alongside marginal validity and predictive efficiency, for CP and indeed any algorithm generating prediction sets. That is, ideally, the validity should hold conditional on any underlying

segmentation of the population. There are alternative forms of conditional validity depending on the type of information that probability of a correct prediction is conditioned on. Vovk (2013) identifies (1) *label* conditional validity for conditioning on the label, (2) *object* conditional validity if conditioning on the features of each example (as opposed to the label) and (3) *training* conditional validity for conditioning on a property of the training set. We can also consider combinations of these types, so if we are interested in object and label conditional validity together, we call this *example* conditional validity. Additionally, CP can suffer from prediction sets of different sizes having different success rates. So, for some applications, smaller prediction sets can have lower success rate. This is also a special case of violation of conditional validity, since in this case the success rate is conditional on prediction set size. We will refer to it as *prediction set size conditional validity*. Since the prediction set is a function of the example, by appying the trained CP to it, it is actually a special case of example conditional validity.

Lei and Wasserman (2014) construct a CP which has asymptotic object conditional validity and asymptotic efficiency in the number of examples used in development (more precisely: in the calibration set), basing their CM on a conditional kernel density estimator. However, for practical purposes and in this article we are interested in finite sample results. McCullagh et al. (2009) present provably conditionally valid prediction intervals, for the finite sample case, based specifically on linear regression, although other parametric models could be used, assuming some known probability distribution. However, it has been shown that it is impossible to meet the criterion of conditional validity in the finite sample case for non-trivial problems in the general case (Vovk, 2013; Lei and Wasserman, 2014). Both papers suggest label-conditioned CPs for classification as a way to address this problem. More generally, a conditional CP can be used that allows the user to specify a taxonomy of data into finite categories and then the CP is valid within those pre-defined categories (Vovk, 2014). However, these approaches are problematic since (1) they limit the conditional validity to just some finite pre-defined categories and do not apply generally to *any* taxonomy, and (2) the effectiveness of the CP is reduced with the more categories included in the taxonomy, since the conditional CP essentially relies on a separate CP for each segment of data, limiting these approaches for practical use with large taxonomies.

These practical limitations with conditional validity have led Barber et al. (2020) to proprose relaxing the requirement for conditional validity and to explore using an *approximation* to conditional validity. They set up a framework called *distribution-free approximate conditional coverage* meaning that conditional validity should be true for at least some predefined proportion of the population, which is similar to the PAC-type conditional validity given by Vovk (2014). However, in the general case, they find that it is also impossible to meet this broad target. They make further progress proposing *restricted conditional coverage* to provide local conditional validity in the neighbourhood (ball) around some point. This is a promising approach and they show that CP satisfies this condition, but only for the case of CP for regression with the standard CM. Additionally, they identify computational difficulties with this method that require further research. The development of the normalized CM for regression (Papadopoulos et al., 2002) is an approach which can be viewed as addressing this problem since the normalization factor allows more difficult prediction cases to be given wider prediction intervals, hence increasing their probability of being correct, and vice versa for easier prediction cases. However, it is a heuristic approach

since it does not directly optimize for approximation to conditional validity and there has been little investigation of its effectiveness for approximation to conditional validity.

It may be supposed that pursuing the goal of maximizing predictive efficiency may align with conditional validity, in some sense. However, Example 3 gives a simple case demonstrating predictive efficiency is not consistent with conditional validity [2].

In this paper, we focus on object conditional validity. The medical and business examples given above are both examples of this. We demonstrate that lack of object conditional validity is a genuine problem for CP through experiments with three data sets and alternative CPs; and, secondly, propose a new *iterative feedback-adjusted conformity measure* (IFACM) algorithm that can be used as part of an inductive conformal predictor (ICP) to achieve approximation to object conditional validity (AOCV), although in a different sense to Barber et al. (2020). We provide some theory and experimental results. The methodology is based on ICP since ICP is a practical version of CP appropriate for batch machine learning. The proposed IFACM algorithm works from the proposition that (1) the AOCV of an ICP can be estimated using a meta-model, and (2) this meta-model can then be used to adjust the CM thus forming a new ICP which is able to achieve better AOCV on the same problem. This process can be repeated until no improvement in measured AOCV can be acheived. Finally, this algorithm will output a CM formed as layers of adjustments to some base CM. The algorithm is similar in approach to that of boosting algorithms that are used to reduce errors on point predictions through layers of simple predictive models (Freund and Schapire, 1999).

The remainder of this article is organized as follows. Section 2 introduces CP and the proposed IFACM algorithm, Section 3 presents some experimental results for both regression and classification, demonstrating the effectiveness of the proposed algorithm to achieve AOCV and conclusions are given in Section 4.

Abbreviations used in this article are listed in Table 1 for the convenience of the reader.

AOCV	Approximation to object conditional validity
CM	Conformity measure
CP	Conformal predictor
DCV	Deviation from conditional validity
ICP	Inductive conformal predictor
IFACM	Iterative feedback-adjusted conformity measure
SR	Success rate

Table 1: Abbreviations used in this article.

2. Thanks to the anonymous reviewer for pointing out that Example 3 is a variation of Cox's example of two measuring instruments (Cox and Hinkley, 1974) adapted to conformal prediction.

Example 1 *Suppose a disease subtype needs to be diagnosed and it must be one of A, B or C. If the predictor outputs the prediction set $\{A, B, C\}$ then it provides no new information at all, since the prediction set just states that the subtype is any of those available. The most informative prediction is the prediction singleton set, say $\{B\}$, since this isolates one single case. The prediction set $\{A, B\}$ offers some information (by excluding C) but is not as informative as the singleton set. Hence this example suggests that cardinality of the prediction set is a good measure of predictive inefficiency.*

Example 2 *Suppose property price is predicted and a prediction interval $[230, 534]$ (in '000 GBP) is given. Then this range is rather broad and may not be particularly useful to a real estate agent. If the prediction algorithm is improved to give prediction interval $[240, 350]$ then this may prove an informative and useful prediction. Hence, this example suggests that width of prediction interval would be a good measure of predictive inefficiency. For these particular examples, the predictive inefficiency would be 304 and 110 respectively.*

Example 3 *A simple example is constructed to demonstrate that finding the most efficient prediction intervals, on average, whilst maintaining marginal validity, leads to conditionally invalid prediction intervals.*

Suppose we have one predictor variable $X \in \{1, 2\}$ with $\mathbb{P}(X = 1) = \frac{1}{2}$, and a label that is uniformly distributed, $Y \sim U(0, X)$. A simple predictor makes prediction intervals $R = [0, a]$ if $X = 1$ and $R = [0, b]$ if $X = 2$, so parameters a and b need to be computed. A significance level $\varepsilon < \frac{1}{2}$ is preset for marginal validity, i.e. $\mathbb{P}(Y \in R) = 1 - \varepsilon$.

By the law of total probability,

$$\begin{aligned}\mathbb{P}(Y \in R) &= \mathbb{P}(Y \in [0, a] \,|X = 1)\mathbb{P}(X = 1) + \mathbb{P}(Y \in [0, b] \,|X = 2)\mathbb{P}(X = 2) \\ &= \mathbb{P}(Y \in [0, a] \,|X = 1) \times \tfrac{1}{2} + \mathbb{P}(Y \in [0, b] \,|X = 2) \times \tfrac{1}{2} \\ &= \tfrac{1}{2}a + \tfrac{1}{4}b = 1 - \varepsilon.\end{aligned}$$

Measure inefficiency as width of the prediction interval, then minimizing mean inefficiency is minimizing

$$\mathbb{E}(|R|) = \mathbb{E}(|R| \,|X = 1)\mathbb{P}(X = 1) + \mathbb{E}(|R| \,|X = 2)\mathbb{P}(X = 2) = \frac{1}{2}a + \frac{1}{2}b.$$

With the marginal validity constraint above, this is a simple linear programming problem with the solution,

$$a = 1, \quad b = 2 - 4\varepsilon,$$

noticing that since $\varepsilon < \frac{1}{2}$, $b > 0$ (an alternative is to use the square of prediction interval width as an inefficiency measure, in which case $\frac{1}{2}a^2 + \frac{1}{2}b^2$ is minimized; however, this

leads to the same solution). This then leads to the two conditional probabilities of being correct:

$$\mathbb{P}(Y \in [0, a] \,|X = 1) = 1, \quad \mathbb{P}(Y \in [0, b] \,|X = 2) = 1 - 2\varepsilon$$

which both deviate from the confidence level $1 - \varepsilon$*, hence the solution does not exhibit conditional validity.*

It is possible to enforce object conditional validity for this simple example by setting $a = 1 - \varepsilon$ *and* $b = 2(1 - \varepsilon)$ *so that*

$$\mathbb{P}(Y \in [0, a] \,|X = 1) = \mathbb{P}(Y \in [0, b] \,|X = 2) = 1 - \varepsilon$$

However, mean inefficiency then increases from $\frac{3}{2} - 2\varepsilon$ *to* $\frac{3}{2}(1 - \varepsilon)$*, a difference of* $\frac{1}{2}\varepsilon$ *more than optimal inefficiency. Therefore enforcing object conditional validity leads to a reduction in predictive efficiency.*

2. Methodology

We first introduce ICP and marginal validity, then discuss AOCV before introducing the IFACM algorithm. Finally we show that IFACM for regression is consistent with the normalized CM.

2.1. Inductive Conformal Prediction

We set up the ICP framework following Vovk et al. (2005).

- Let $\boldsymbol{z} = (\boldsymbol{x}, y)$ denote examples from the same unknown distribution P such that $\boldsymbol{x}$ is a vector of m predictor variables $\boldsymbol{x} \in \mathbb{R}^m$ and label $y \in \mathbb{Y}$ for some set of labels $\mathbb{Y}$.
- Let $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_l$ be a particular sequence of examples $\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j)$ from P.
- Let 1 to k index the training data set and $k + 1$ to l index the calibration set, for $1 < k < l$.
- A conformity measure (CM) is a function

  $$A(\boldsymbol{z}) = \mathcal{A}(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k, \boldsymbol{z})$$

  such that $\mathcal{A}$ is exchangeable: ie $\mathcal{A}(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k, \boldsymbol{z}) = \mathcal{A}(\boldsymbol{z}_{\pi(1)}, \cdots, \boldsymbol{z}_{\pi(k)}, \boldsymbol{z})$ for all permutations π of $1, \cdots, k$. For convenience, we will sometimes use $A(\boldsymbol{x}, y)$ to denote $A(\boldsymbol{z})$.
- Let ε be a preset significance level, so $1 - \varepsilon$ is the confidence level for predictions.

We typically consider a CM having the form $A(\boldsymbol{z}) = \mathcal{A}^p(\theta, \boldsymbol{z})$ for some parameters $\theta = M(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k)$ where M is intended to be a model structure within which θ are estimated based on data $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k$. For example, for regression, M could be ordinary least squares

(OLS) linear regression and θ, the vector of coefficients in the OLS linear regression model; or M could be an artificial neural network with θ the vector of weights in the network.

The ICP is defined as the prediction algorithm that gives the prediction set at significance level ε for a new example $\boldsymbol{x}$, based on the calibration set of examples $k+1$ to l as

$$\Gamma^{\varepsilon}(\boldsymbol{x}; A) = \left\{ y \in \mathbb{Y} : \sum_{j=k+1}^{l} \mathbb{I}\left[A(\boldsymbol{x}, y) \geq A(\boldsymbol{z}_j)\right] + 1 > \varepsilon(l-k+1) \right\} \tag{1}$$

where $\mathbb{I}$ is the usual indicator function. This prediction set is dependent on the CM used and this is made explicit on the left-hand side of the expression. For a new example $(\boldsymbol{x}_i, y_i)$, the prediction set is *correct* if the true label is an element; i.e. $y_i \in \Gamma^{\varepsilon}(\boldsymbol{x}_i; A)$. Assuming only that the calibration data set, $\boldsymbol{z}_{k+1}, \cdots, \boldsymbol{z}_l$, and any new test or operational data sets are exchangeable, ICP predictions on the new data are marginally valid [3] :

$$\mathbb{P}(y_i \in \Gamma^{\varepsilon}(\boldsymbol{x}_i; A)) = 1 - \varepsilon. \tag{2}$$

When the label set $\mathbb{Y}$ is finite, this is ICP for classification. When $\mathbb{Y} = \mathbb{R}$, this is ICP for regression. For these two cases, we can consider two particular CMs that are common in the literature.

Scoring CM A scoring function $f : \mathbb{R}^m \times \mathbb{Y} \to \mathbb{R}$ for which a higher output of $f(\boldsymbol{x}, y)$ means that label y is (in some sense) more likely to be associated with example $\boldsymbol{x}$, then f itself can be used directly as a CM. An example of this is multinomial logistic regression for classification which estimates coefficient vectors β_y for each possible label $y \in \mathbb{Y}$ such that $f(\boldsymbol{x}, y) = \beta_y \cdot \boldsymbol{x}$ is the log-odds of y conditional on $\boldsymbol{x}$. We will use the multinomial logistic regression scoring function in the later experimental section.

Normalized CM In the case of regression, a typical choice is the normalized nonconformity measure:

$$N(\boldsymbol{x}, y) = \frac{|y - \hat{y}|}{\hat{\sigma}} \tag{3}$$

where $\hat{y}$ is a point estimate based on a regression model for y and $\hat{\sigma} > 0$ is some estimate of uncertainty in the prediction given by the model (Papadopoulos et al., 2002). Typically $\hat{\sigma}$ is also modelled using a regression model; i.e. estimated on the residuals given by the regression model for y on the training data, $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k$. For later experiments in this article, OLS linear regression of y is used for the numerator and a OLS linear regression of the log of absolute residual is used for σ following Papadopoulos et al. (2002). For consistency, we use CM for this study and a normalized CM can be constructed as the reciprocal of the normalized nonconformity measure $A_{\mathrm{N}}(\boldsymbol{x}, y) = 1/N(\boldsymbol{x}, y)$. For the normalized nonconformity measure

$$\Gamma^{\varepsilon}(\boldsymbol{x}; A_{\mathrm{N}}) = [\hat{y} - \hat{\sigma}q,\ \hat{y} + \hat{\sigma}q] \tag{4}$$

3. Strictly speaking, exact validity is only true for the smoothed CP (Vovk et al., 2005). In the case when CM generates ties, we would have conservative validity: $\mathbb{P}(y \in \Gamma^{\varepsilon}(\boldsymbol{x})) \geq 1 - \varepsilon$. However, for practical purposes with sufficient granularity in the CM, exact validity is approximately the case for ICP.

thus forming a prediction interval, where q is the ε-quantile of the empirical distribution of CM values within the calibration set. In the special case when $\hat{\sigma} = 1$, this gives the *standard CM*. As can be seen from (4), the standard CM always gives the same fixed width $2q$ for prediction intervals. The introduction of the normalizing denominator allows for variation in the prediction interval width and hence form a CP that has varied prediction interval widths according to the difficulty of the observation, with observations that are more difficult to predict having wider prediction intervals.

2.2. Approximation to Object Conditional Validity

As discussed earlier, it would be valuable to achieve object conditional validity given by

$$\mathbb{P}(y \in \Gamma^{\varepsilon}(\boldsymbol{x}; A) \mid \boldsymbol{x}) = 1 - \varepsilon$$

for all $\boldsymbol{x} \in \mathbb{R}^m$, but, in the nontrivial case, this is impossible to achieve for CPs as shown by Vovk (2013) and Lei and Wasserman (2014). Instead we consider relaxing the constraint and allow for AOCV given broadly by

$$\mathbb{P}(y \in \Gamma^{\varepsilon}(\boldsymbol{x}; A) \mid \boldsymbol{x}) \approx 1 - \varepsilon.$$

This requirement can be made more precise by expressing it as minimizing the square difference across the population,

$$\mathbb{E}_{\boldsymbol{x}}\left[\left(\mathbb{P}(y \in \Gamma^{\varepsilon}(\boldsymbol{x}; A) \mid \boldsymbol{x}) - (1 - \varepsilon)\right)^2\right]. \tag{5}$$

This approach to AOCV is different to Barber et al. (2020) who, instead, consider proportions of the population that are exactly conditionally valid as an approximation method. After conformal predictions are made on a data set indexed by some index set S, (5) is estimated as

$$\frac{1}{|S|}\sum_{j \in S}\left(p(\boldsymbol{z}_j; A) - (1 - \varepsilon)\right)^2$$

where

$$p((\boldsymbol{x}, y); A) = \hat{\mathbb{P}}(y \in \Gamma^{\varepsilon}(\boldsymbol{x}; A) \mid \boldsymbol{x}) \tag{6}$$

is some probability estimator. The probability estimator can be logistic regression, k-nearest neighbours (k-NN) or any other probabilistic model taking training data $(\boldsymbol{x}_j, c_j)\ \forall j \in S$ where

$$c_j = \mathbb{I}\left[y_j \in \Gamma^{\varepsilon}(\boldsymbol{x}_j; A)\right]$$

indicates whether the prediction set for example j is correct. Therefore, this measure can be used to quantify AOCV. We take the square root of this estimate, so that the measure is on the same scale as the conditional probability. Hence the root mean square error gives *deviation from conditional validity* (DCV),

$$\mathrm{DCV}(A) = \sqrt{\frac{1}{|S|}\sum_{j \in S}\left(p(\boldsymbol{z}_j; A) - (1 - \varepsilon)\right)^2}.$$

If an ICP is conditionally valid then DCV(A) ≈ 0. The closer it is to zero, the better the AOCV exhibited by the ICP. Essentially $p(\boldsymbol{z}; A) \approx 1-\varepsilon$ indicating an uninformative model. For example, using logistic regression for p, the coefficient estimates will be close to zero and statistically insignificant.

On the other hand, a low value of DCV does not *guarantee* conditional validity, since it only reflects the model structure that is used to construct the probability estimate, such as logistic regression, and not all possible conditionalities in the data. However, it will be a good indicator of AOCV for practical purposes.

2.3. Iterative Feedback-adjusted Conformity Measure

Consider constructing CMs that seek to minimize DCV(A). Our proposal is to extend an existing base CM A which may have a high DCV with another as follows:

$$A^*_\gamma(\boldsymbol{z}) = A(\boldsymbol{z}) + \gamma U(\boldsymbol{z}; A)$$

where $\gamma > 0$ is a control parameter, U is an update function based on the discrepency between estimated probability given A and confidence level, with the properties,

- $U(\boldsymbol{z}; A) \geq 0$ if $p(\boldsymbol{z}; A) < 1-\varepsilon$;
- $U(\boldsymbol{z}; A) = 0$ if $p(\boldsymbol{z}; A) = 1-\varepsilon$;
- $U(\boldsymbol{z}; A) \leq 0$ if $p(\boldsymbol{z}; A) > 1-\varepsilon$.

where the underlying estimator p is exchangeable, i.e. the estimates are invariant to the order of the training examples. This ensures that A^*_γ is also a CM. This exchangeability condition is true of most probabilistic estimators and, in particular, is true of logistic regression and k-NN where the order of training examples is irrelevant. The inituition for the update function is that if the conditional probability of being correct is too low, then we increase the value of the CM so that the correct label has more chance of being in the prediction interval (1), hence increasing the conditional probability of being correct if ICP is rerun with the updated CM, and vice versa, for conditional probabilities that are too high, the CM value is lowered following the update, thus making it less likely that the prediction is correct.

A specific update function considered in this study is

$$U^*(\boldsymbol{z}; A, \delta) = \begin{cases} +1 & \text{if } p(\boldsymbol{z}; A) < (1-\varepsilon) - \delta \\ -1 & \text{if } p(\boldsymbol{z}; A) > (1-\varepsilon) + \delta \\ 0 & \text{otherwise} \end{cases}$$

where $\delta \geq 0$ is a sensitivity parameter. Notice that U^* is not dependent on y but this is not a problem, since A is expected to provide discrimatory power for the ICP, whilst U^* just provides an adjustment to lead the ICP to achieve better AOCV. Ideally we would like to prove that adjusting by U^* will improve AOCV in some way: perhaps showing that DCV is reduced. However, this seems difficult to achieve, at least in the general case, and remains an area for further research. However, it is possible to prove that applying U^* adjustment to the CM does improve the average log-odds of being correct for those

observations whose outcome (i.e. c_j) is switched by the adjustment, so that it is closer to the target confidence level, in the case when logistic regression is used as the probability estimator p. Technical details are provided in online supplementary material or available on request from the author.

The updated CM A^*_γ is optimized with respect to γ and δ on the training data set with the goal to minimize the DCV measure. There is no gradient that can be used for optimization but with only two parameters to minimize across, Nelder-Mead optimization (Nelder and Mead, 1965) is adequate for this task. Since both parameters are greater or equal to zero, the optimization is performed over the logarithm of the parameters.

The U^* update function is rather simplistic and one application of an update may provide just a small improvement in DCV. However, since the update A^* is itself a CM, the application of the update function can be applied iteratively, minimizing DCV at each iteration, until it cannot be minimized any further. Representing the CM recursively:

$$A^{*[i]}(\boldsymbol{z}) = \begin{cases} A(\boldsymbol{z}) & \text{if } i = 0, \\ A^{*[i-1]}(\boldsymbol{z}) + \gamma_i U^*(\boldsymbol{z}; A^{*[i-1]}, \delta_i) & \text{if } i > 0 \end{cases}$$

for parameters $\gamma_i > 0$ and $\delta_i \geq 0$ that are computed at each iteration to minimize DCV. Importantly, the probability estimator (6) is re-estimated at each iteration based on the ICP with the current version of $A^{*[i]}$. Then the output after some n_u iterations is the IFACM expressed as the weighted sum,

$$A^{**}(\boldsymbol{z}) = A(\boldsymbol{z}) + \sum_{i=1}^{n_u} \gamma_i U^*(\boldsymbol{z}; A^{*[i-1]}, \delta_i). \tag{7}$$

This use of iterative modifications to the CM based on feedback from previous stages of the ICP performance mirrors the method of boosting that is used to reduce errors iteratively by giving greater weight to examples with errors from previous iterations in each model build at the next iteration. We would expect that the process of adjusting the CM by iterative updates will lead to reduced DCV, but on the other hand, that there will be a trade-off in an increase in predictive inefficiency. Hence, to mitigate against this the optimization task is modified to minimize at each iteration i,

$$\text{DCV}(A^{*[i]}) + C(I(A^{*[i]}) - I(A))\mathbb{I}\left[I(A^{*[i]}) > I(A)\right] \tag{8}$$

where I is an inefficiency measure for predictions made using the CM on index set S and $C \geq 0$ is a control hyperparameter. There are several alternative measures of prediction inefficiency (Vovk et al., 2016). In this study, the mean size of prediction sets is used,

$$I(A') = \frac{1}{|S|} \sum_{j \in S} \left|\Gamma^\varepsilon(\boldsymbol{x}_j; A')\right|. \tag{9}$$

For classification, the size is the cardinality of the prediction set. For regression, using normalized CM, the size is the width of the prediction interval.

2.4. Pseudo-ICP

The method is introduced with the abstract set S. In a pure ICP setting, S would be a different data set at each iteration of IFACM update, to ensure independence between calibration and test sets at each stage. This would ensure that the ICP used at each stage is valid. However, this is not a practical use of data, unless a huge amount of data is available. It is also unnecessary, since the procedure does not need to be valid for each iteration of the update to IFACM. As long as the IFACM output is used within a proper ICP at the final stage, with independent calibration and test sets, the final output will indeed be valid. To this end, we introduce the notion of a *pseudo-ICP* which is the same algorithm as ICP except that the same training data set is used for both calibration and test. Formally,

$$\Gamma^{\varepsilon}_{\text{pseudo}}(\boldsymbol{x}; A) = \left\{ y \in \mathbb{Y} : \sum_{j=1}^{k} \mathbb{I}\left[A(\boldsymbol{x}, y) \geq A(\boldsymbol{z}_j)\right] + 1 > \varepsilon k \right\}.$$

This pseudo-ICP mimics ICP sufficiently to be used to optimize IFACM for following use in a proper ICP. This approach follows the use of a surrogate CP in Bellotti (2020). We anticipate that the multiple use of one data set for training, calibration and test set could lead to overfitting, and hence lead to undesirable outcomes, so this will be checked as part of the experimental results.

2.5. IFACM algorithms

Algorithms 1 and 2 show how to compute the IFACM using the pseudo-ICP. Note that these algorithms are a prelude to the ICP. Once they have computed the IFACM, this is passed on to be used as the CM within a genuine ICP with an independent calibration data set and which is therefore valid.

Algorithm 1: Build pseudo-ICP on training data to measure DCV

input : *Train* = training data set,
CL = confidence level,
A = base conformity measure.

Compute prediction intervals *PI* for all observations in *Train* using ICP with A, *CL* and *Train* as the calibration data set;
Construct a probabilistic model p of the correctness of *PI* on *Train*, i.e. formula (6);
Compute *DCV* using p predicting on *Train*, relative to *CL* ;
$W \leftarrow$ mean width of *PI*, using (9) ;
output: DCV, W

2.6. IFACM and Normalized CM

The CM A^{**} is consistent with the normalized CM for regression. To see this, first observe that applying a monotonically increasing transform to a CM does not change the behaviour of the ICP since it is the rank ordering of CM that matters, as can be seen in (1). Therefore,

Algorithm 2: Iterative feedback-adjusted conformity measure (IFACM)

input : *Train* = training data set,
CL = confidence level,
C = control parameter in equation (8),
B = Base conformity measure.

$(DCV_1, W_0) \leftarrow$ Algorithm 1 with input: *Train*, *CL*, B;
$A \leftarrow B$;
$i \leftarrow 1$;
repeat
  Run Algorithm 1 multiple times with input: *Train*, *CL*, $A + \gamma_i U^*(.; A, \delta_i)$ to minimize $DCV + C(W - W_0)\mathbb{I}(W > W_0)$ with respect to γ_i, δ_i using the Nelder-Mead algorithm (*i.e. minimize (8) with respect to* γ_i, δ_i) ;
  $DCV_2 \leftarrow DCV$ from the minimization step above ;
  if $DCV_2 < DCV_1$ **then**
    $DCV_1 \leftarrow DCV_2$;
    $A \leftarrow A + \gamma_i U^*(.; A, \delta_i)$;
    $i \leftarrow i + 1$
  end
until $DCV_2 \geq DCV_1$;
$n_u \leftarrow i$;
output: A

taking base $A(\boldsymbol{x}, y) = \log \sigma - \log |y - \hat{y}|$ and applying the exponential function to (7) gives

$$\exp A^{**}(\boldsymbol{x}, y) = \frac{\hat{\sigma}}{|y - \hat{y}|} \exp \left[\sum_{i=1}^{n_u} \gamma_i U^*(\boldsymbol{x}, y; A^{*[i-1]}, \delta_i) \right].$$

Since the exponential function is monotonically increasing, this CM will give the same ICP as (7), except in the transformed version it can be seen that this is the normalized CM with an extra product update. Indeed, the update functions can be interpreted as a further normalizing factor in the CM. Therefore from (4), the prediction interval yielded by this CM for regression is

$$\Gamma^{\varepsilon}(\boldsymbol{x}; A^{**}) = \left[\hat{y} - \hat{\sigma}' q,\ \hat{y} + \hat{\sigma}' q \right] \tag{10}$$

where

$$\hat{\sigma}' = \hat{\sigma} \exp \left[\sum_{i=1}^{n_u} \gamma_i U^*(\boldsymbol{x}, y; A^{*[i-1]}, \delta_i) \right].$$

3. Experimental settings and results

Experiments were conducted with three data sets listed in Table 2. Data set "covtype" has outcome identifying tree cover type for different forest locations. It is used in this study since it is a large multi-class classification problem. Data set "GPU" provides GPU performance data for different GPU settings, outcome is average performance time and it

is a large regression data set (Nugteren and Codreanu, 2012) . Data set "KC" (King's County) contains US housing data and the outcome variable is property sale price. It is a smaller regression problem but is interesting for this study since the normalized CM gives improved performance over the standard CM, which was not the case with the GPU regression problem. Both "covtype" and "GPU" are available from the UCI Machine Learning data repository (Frank and Asuncion, 2010) and "KC" is available from the Kaggle website (`www.kaggle.com`).

Data set	#var	Outcome (#labels)	#Train	#Cal	#Test
covtype	54	Classification (7)	100000	100000	100000
GPU	14	Regression	20000	20000	20000
KC	24	Regression	10000	5000	6613

Table 2: Data sets. #var = number of predictor variables, #labels = number of unique class labels for classification, #Train, #Cal, #Test are number of examples in the training, calibration and test sets, respectively.

All data sets were randomly partitioned into three parts for training, calibration and testing, according to the number of examples shown in Table 2. As discussed in Section 2.4, only the training data is used to construct the IFACM and this is then used within a ICP taking the independent calibration data set and independent test data set for evaluation.

Parameter settings are as follows:-

- *Confidence level.* For all experiments, confidence level is set to make the problem sufficiently hard that predictive efficiency becomes a challenge. For "covtype" this is 0.95 and for "GPU" and "KC" it is 0.9. Additionally a confidence level of 0.99 is also used for "GPU" to explore the consequences of high confidence levels.
- *Base CM.* For classification, the multinomial scoring CM is used as base CM, whereas normalized CM is used for the regression problems. Additionally standard CM for regression is used for "KC" to explore if normalized CM improves on standard CM in terms of AOCV. Both use a base OLS linear regression model for point estimates of $\hat{y}$ and normalized CM also uses OLS linear regression for estimates of $\log \hat{\sigma}$ for (3).
- *Probability estimator.* Logistic regression is used as the probability estimator p (6).
- *C: penalty hyperparameter in (8).* For "covtype", it was possible to achieve good results with $C = 0$. However, for the other data sets, $C = 0$ leads to high predictive inefficiency, hence a value of $C = 0.5$ is used instead. For contrast, $C = 5$ is also used.
- *Nelder-Mead optimization.* Low starting values $\log \gamma_i = -5$, $\log \delta_i = -5$ were used when minimizing (8).

The key performance measure is DCV for these experiments, but inefficiency is also a performance measure and (9) is used to compute this. Additionally, the validity of CP needs to be checked, so success rate (SR) defined as

$$\text{SR} = \frac{1}{|S|} \sum_{j \in S} c_j$$

is reported for test set S and we expect SR to be approximately equal to CL.

Results are shown in Table 3. This shows the main result that for all data sets used, both regression and classification, the IFACM algorithm reduces DCV on both the training and test data, whilst maintaining marginal validity (SR $\approx$ CL), without a large increase in predictive inefficiency. Therefore, IFACM behaves as designed and demonstrates an improvement in AOCV. There are other important observations, as follows.

- As expected, the reduction in DCV comes at the cost of a small increase in predictive inefficiency and this is controlled by the hyperparameter C. However, the impact of this trade-off is different for each data set. For "covtype", inefficiency increases from 1.78 to 1.84 for the best improvement in DCV (when $C = 0$, lines 3–4), whereas taking $C = 5$ will yield better inefficiency (1.80) but the DCV more than doubles, so the impact of the trade-off between DCV and inefficiency is high for this data set. However, for "GPU", a low DCV is measured with minimal increase in inefficiency when C is low (lines 11–12). Surprisingly, when a high confidence level is used, IFACM yields lower inefficiency even with low DCV (lines 15–18). Whether this is a general result requires further investigation, but at the very least it demonstrates that for "GPU", the impact of applying IFACM on inefficiency is minimal.

- In many cases, DCV is much lower on the training data set compared to the test set. This is especially the case when low values of C are used (compare line 11 to 12 and 25 to 26). This suggests that in some situations, the IFACM algorithm gives rise to a degree of overfit for the goal of minimizing DCV. For the data sets used in the study, it does not give rise to a significant problem with performance on the test data; however, it is worth some further investigation.

- Lines 19–22 demonstrate that for the "KC" regression problem, using normalized CM gives lower predictive inefficiency than the standard CM which confirms what we expect from previous work, e.g. (Papadopoulos et al., 2002), but also interestingly the normalized CM is able to reduce DCV on training and test data (lines 21–22), which is expected by construction of the normalized CM. Nevertheless, we see that applying IFACM with the normalized CM as base CM allows for even further reduction in DCV. Using $C = 5$, IFACM is able to achieve this with only a small increase in inefficiency (lines 29–30). Interestingly, applying IFACM on the standard CM gives improvement over standard CM, but is not as good as IFACM applied to normalized CM in terms of either inefficiency or DCV measures (compare lines 27–28 to 29–30). This is important since it shows that IFACM cannot simply replace a good base CM (i.e. the normalized CM) but is a supplement to an existing base CM, for the purpose of achieving AOCV.

We can look in further detail at the behaviour of the IFACM algorithm.

- Figure 1 shows the coefficient estimates for each of the variables in "GPU" using the logistic regression probability estimator p in (6), before and after IFACM is applied, corresponding to lines 10 and 12 of Table 3 respectively. Since data has been standardized, the magnitude of the coefficients are comparable. The left graph shows the result on training data: in this case, the reduction in magnitude of coefficients is large,

#	Data set	Segment	Algorithm	CL	C	SR	Ineff.	DCV
1	covtype	training	Base ICP	0.95	-	0.951	1.78	0.0396
2		test	Base ICP	0.95	-	0.950	1.78	0.0395
3		training	IFACM	0.95	0	0.951	1.84	0.00345
4		test	IFACM	0.95	0	0.950	1.84	0.00705
5		training	IFACM	0.95	1	0.950	1.82	0.00953
6		test	IFACM	0.95	1	0.949	1.83	0.0101
7		training	IFACM	0.95	5	0.950	1.80	0.0193
8		test	IFACM	0.95	5	0.949	1.80	0.0185
9	GPU	training	Base ICP / NCM	0.9	-	0.901	2.20	0.0497
10		test	Base ICP / NCM	0.9	-	0.898	2.20	0.0503
11		training	IFACM / NCM	0.9	0.5	0.901	2.23	0.000949
12		test	IFACM / NCM	0.9	0.5	0.898	2.23	0.00725
13		training	IFACM / NCM	0.9	5	0.901	2.20	0.00497
14		test	IFACM / NCM	0.9	5	0.901	2.20	0.00954
15	GPU	training	Base ICP / NCM	0.99	-	0.991	3.38	0.0162
16		test	Base ICP / NCM	0.99	-	0.991	3.38	0.0181
17		training	IFACM / NCM	0.99	0.5	0.991	3.27	0.00117
18		test	IFACM / NCM	0.99	0.5	0.991	3.28	0.00260
19	KC	training	Base ICP / SCM	0.9	-	0.908	1.58	0.121
20		test	Base ICP / SCM	0.9	-	0.900	1.58	0.135
21		training	Base ICP / NCM	0.9	-	0.903	1.49	0.0549
22		test	Base ICP / NCM	0.9	-	0.900	1.49	0.0587
23		training	IFACM / SCM	0.9	0.5	0.900	1.70	0.0359
24		test	IFACM / SCM	0.9	0.5	0.900	1.69	0.0316
25		training	IFACM / NCM	0.9	0.5	0.903	1.55	0.00200
26		test	IFACM / NCM	0.9	0.5	0.900	1.55	0.0181
27		training	IFACM / SCM	0.9	5	0.901	1.60	0.0475
28		test	IFACM / SCM	0.9	5	0.900	1.59	0.0455
29		training	IFACM / NCM	0.9	5	0.902	1.52	0.0162
30		test	IFACM / NCM	0.9	5	0.900	1.52	0.0199

Table 3: Experimental results. SCM=standard CM, NCM=normalized CM, CL=confidence level $1 - \varepsilon$, C=penalty hyperparameter in (8), Ineff.=mean prediction interval width.

leading to an uninformative model after IFACM (i.e. coefficients ≈ 0) [4]. The right graph shows coefficients on test data: overall, there is a reduction in magnitude, but it is not as pronounced as with training data and some coefficients actually become larger in magnitude. This demonstrates there may be overfit on the training data, i.e. it does not perform so well on test data as training, in terms of AOCV, corroborating

4. For this problem, notice that we are in the peculiar position of wanting to achieve the *worst* model in the sense of weak model fit, so that there is no relationship between whether a prediction set for an object $\boldsymbol{x}$ is correct and $\boldsymbol{x}$ itself.

the results for DCV given in Table 3. However, as discussed above, this does not lead to a significant degradation in test performance.

- Table 4 shows the sequence of values of parameters γ_i and δ_i estimated in several experiments. They show that the number of iterations of the IFACM algorithm n_u is not large, with maximum 5 in our experiments. Also, the number of iterations goes down as C increases, showing that greater control of inefficiency means less precision with the feedback adjustment. The sequence of values of γ_i decreases with each iteration, as the magnitude of the adjustment at each iteration is reduced. There is no obvious pattern with the sequences of δ_i values, however.

- The DCV reported in Table 3 measures average deviation from conditional validity across the whole data set and demonstrates success in reducing this. Nevertheless, to convince ourselves that the method is genuinely effective, it is also valuable to show SR for different sub-segments of the data. Therefore, as an illustration, Table 5 shows this for several segments in the test data sets. These segments are chosen as deviating greatly from the confidence level when the Base ICP is used. The results show that for each of these segments, using IFACM makes a considerable difference, giving SR much closer to the target confidence level, within each segment. So, e.g. in "covtype", with the Base ICP, for V43 we can see a bias in SR with better performance for V43=1 (SR=0.973). However, once IFACM is used, this bias is reduced and SR for V43=1 becomes closer to the target confidence level 0.95 (SR=0.955). Although improvements in AOCV are observed within all segments, some segments receive greater improvement than others. For example, in "GPU", the improved AOCV is much better for KWG than for the segment MWG=64.

The code was written in R and was run on a PC with a processor running at 2.1 GHz with 16GB RAM. The Nelder-Mead optimizer was set to take no more than 100 steps each iteration of the IFACM algorithm. The computation times for training in Algorithm 2 on "covtype", "GPU" and "KC" data sets (corresponding to lines 3, 11 and 25 of Table 3, respectively) were 6125, 225 and 51 seconds respectively. The much longer running time for "covtype" was a consequence of a larger training data set size and larger number of iterations, n_u.

4. Conclusion

The IFACM algorithm is introduced as a mechanism to iteratively update a base CM in order to adjust the conditional probability of a predictive set being correct so that it is close to the target confidence level across the population and hence achieve AOCV. This study demonstrates IFACM as a proof of concept, and it is applied and studied across three data sets. The experimental results provide good evidence that IFACM algorithm works well at this task, whilst maintaining marginal validity when the IFACM is used in the ICP. There is a cost in terms of increased predictive inefficiency, but the impact of this cost differs by data set and for all the data sets used in this study, the cost is not too onerous with the correct choice of hyperparameter C.

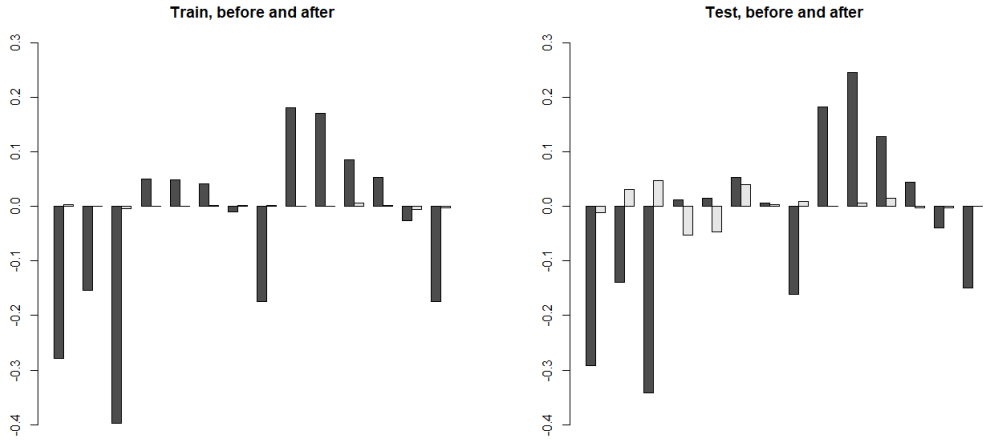

Figure 1: Change in coefficient estimates in the logistic regression probability estimator p before (dark grey) and after (light grey) applying the IFACM algorithm on "GPU" training data (left) and test data (right).

Data set	CL	C		$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
covtype	0.95	0	γ_i	0.644	0.222	0.104	0.086	0.018
			δ_i	0.00002	0.00015	0.00016	0.00318	0.00089
covtype	0.95	1	γ_i	0.574	0.196			
			δ_i	0.00005	0.00016			
covtype	0.95	5	γ_i	0.448				
			δ_i	0.00008				
GPU	0.9	0.5	γ_i	0.157	0.031	0.010		
			δ_i	0.00140	0.00152	0.00063		
GPU	0.9	5	γ_i	0.152	0.026			
			δ_i	0.00103	0.00100			

Table 4: Optimized values of γ_i and δ_i in (7) for each iteration i of Algorithm 2, for different experiments. CL=confidence level $1-\varepsilon$, C=penalty hyperparameter in (8),

Data set	CL	C	Variable (type)	Value	#	SR: Base ICP	SR: IFACM
covtype	0.95	0	V1	<median	50000	0.928	0.945
			(numeric)	≥median	50000	0.972	0.955
			V43	0	80213	0.944	0.949
			(category)	1	19787	0.973	0.955
GPU	0.9	0.5	KWG	16	16212	0.933	0.896
			(category)	32	23788	0.874	0.902
			MWG	16	3097	0.913	0.891
			(category)	32	7823	0.916	0.895
				64	12595	0.916	0.910
				128	16485	0.872	0.896

Table 5: SR within some segments of the test data set using the Base ICP and ICP with IFACM. Each line shows SR measured within just the segment of examples given by the value of the variable. CL=confidence level $1-\varepsilon$, C=penalty hyperparameter in (8), # is the number of examples within the segment.

The results show that with the baseline ICPs, deviation from object conditional validity can be high and hence we recommend that developers using ICP should control for all three objectives:

- *Marginal validity*: guaranteed by (2) for any ICP with independent and identically distributed training, calibration and test (or operational) data;
- *Predictive inefficiency*: needs to be minimized; e.g. minimizing mean prediction interval width (9) for regression;
- *Deviation from conditional validity* (DCV): needs to be minimized; this is the main objective of this study.

This article introduces the problem and proposes the novel IFACM algorithm with some initial experimental results that demonstrate that the algorithm is effective. There remain several avenues for further research.

1. Further benchmark studies on different data sets, different underlying machine learning algorithms for the base CM and alternative U update functions.
2. Exploring the effect of data set size and model complexity for the overfitting problem identified in the experimental results.
3. Testing alternative probability estimators p, such as nonparametric k-nearest neighbours, or estimators that could alleviate the overfitting problem, such as those using regularization.
4. As recommended above, CP should be built taking into account minimizing both predictive inefficiency and DCV. Colombo and Vovk (2020) and Bellotti (2020) propose methods specifically for minimizing predictive inefficiency, whilst this article

focusses just on DCV. It would be interesting to develop a method that integrates both approaches. This is not straightforward since the approaches to optimization are different in each case.

5. This study only considers object conditional validity. It is important to extend to other forms of conditional validity, such as label, example and prediction set size conditional validity.

References

V Balasubramanian, S-S Ho, and V Vovk, editors. *Conformal Predictions for Reliable Machine Learning: Theory, Adaptations and Applications*. Elsevier, 2014.

RF Barber, Candes EJ, Ramdas A, and Tibshirani RJ. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 00: 1–28, 2020.

A Bellotti. Reliable region predictions for automated valuation models. *Annals of Mathematics and Artificial Intelligence*, 81:71–84, 2017.

A Bellotti. Constructing normalized nonconformity measures based on maximizing predictive efficiency. In A Gammerman, V Vovk, Z Luo, E Smirnov, and G Cherubin, editors, *Proceedings of Machine Learning Research: COPA 2020*, volume 128, pages 41–54, 2020.

N Colombo and V Vovk. Training conformal predictors. In A Gammerman, V Vovk, Z Luo, E Smirnov, and G Cherubin, editors, *Proceedings of Machine Learning Research: COPA 2020*, volume 128, pages 55–64, 2020.

DR Cox and DV Hinkley. *Theoretical Statistics*. Chapman and Hall / CRC, 1974.

A Frank and A Asuncion. UCI Machine Learning Repository. *Irvine, CA*, 2010. URL http://archive.ics.uci.edu/ml.

Y Freund and RE Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

A Khosravi, S Nahavandi, D Creighton, and AF Atiya. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Tranactions on Neural Networks*, 22(3), March 2011.

J Lei and L Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society. Series B*, 76:71–96, 2014.

Z Lim and A Bellotti. Normalized nonconformity measures for automated valuation models. *Expert Systems with Applications*, 180:115165, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.115165. URL https://www.sciencedirect.com/science/article/pii/S0957417421006059.

P McCullagh, V Vovk, I Nouretdinov, D Devetyarov, and A Gammerman. Conditional prediction intervals for linear regression. In *Proceedings of the eighth international conference on machine learning and applications*, 2009.

J.A. Nelder and R Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

IR Nouretdinov, A Bellotti, and A Gammerman. Biomedical applications: Diagnostics and prognostics. In V Balasubramanian, S-S Ho, and V Vovk, editors, *Conformal Predictions for Reliable Machine Learning: Theory, Adaptations and Applications*. Elsevier, 2014.

C Nugteren and V Codreanu. CLTune: A Generic Auto-Tuner for OpenCL Kernels. In *MCSoC: 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip. IEEE*, 2012.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive Confidence Machines for Regression. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Machine Learning: ECML 2002. ECML 2002. Lecture Notes in Computer Science*, volume 2430. Springer, Berlin, Heidelberg, 2002.

T Pearce, M Zaki, A Brintrup, and A Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *Proceedings of Machine Learning Research: International Conference on Machine Learning 2018*, 80:4075–4084, 2018.

V Vovk. Conditional validity of inductive conformal predictors. *Machine Learning*, 92: 349–376, 2013.

V Vovk. Beyond the basic conformal prediction framework. In V Balasubramanian, S-S Ho, and V Vovk, editors, *Conformal Predictions for Reliable Machine Learning: Theory, Adaptations and Applications*. Elsevier, 2014.

V Vovk, A Gammerman, and G Shafer. *Algorithmic learning in a random world*. Springer US, 2005.

V Vovk, V Fedorova, I Nouretdinov, and A Gammerman. Criteria of efficiency for conformal prediction. In *COPA 2016: Proceedings of the 5th International Symposium on Conformal and Probabilistic Prediction with Applications*, volume 9653, pages 23–29, April 2016.