Cross-conformal predictive distributions

Vladimir Vovk V.VOVK@RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK
Ilia Nouretdinov I.R.NOURETDINOV@RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK
Valery Manokhin VALERY.MANOKHIN.2015@LIVE.RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK
Alex Gammerman A.GAMMERMAN@RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Ralf Peeters

Abstract

Conformal predictive systems are a recent modification of conformal predictors that output, in regression problems, probability distributions for labels of test observations rather than set predictions. The extra information provided by conformal predictive systems may be useful, e.g., in decision making problems. Conformal predictive systems inherit the relative computational inefficiency of conformal predictors. In this paper we discuss two computationally efficient versions of conformal predictive systems, which we call split conformal predictive systems and cross-conformal predictive systems, and discuss their advantages and limitations.

Keywords: Conformal prediction, cross-conformal prediction, inductive conformal prediction, predictive distributions, split conformal prediction, regression.

1. Introduction

Two sister methods that have been widely presented at the COPA series of workshops are conformal prediction and Venn prediction. Both methods enjoy provable properties of validity under the IID model but their outputs are very different: whereas Venn predictors output probabilities (more precisely, upper and lower probabilities), conformal predictors output p-values (often packaged as prediction sets). Not only the outputs but also the areas of application are different for the two methods: Venn predictors, in their standard form, are only applicable to classification problems[1] whereas conformal predictors are applicable to both classification and regression.

A recent development in conformal prediction has been the definition and study of conformal predictive systems (CPS, which we will use for both singular and plural) in Vovk et al. (2017), based on the parallel work on predictive distributions in parametric statistics (see, e.g., Schweder and Hjort, 2016, Chapter 12, and Shen et al., 2018). In the case of regression problems, CPS output predictive distributions; the difference between p-values and probabilities is often emphasized in statistics, but in the case of CPS the p-values get arranged into a probability distribution thus becoming probabilities, in a sense. This

1. See, however, Nouretdinov et al. (2018) for the latest development.

facilitates new uses of conformal prediction, such as automatic decision making (Vovk and Bendtsen, 2018). However, for many underlying algorithms CPS (like conformal predictors in general) are computationally inefficient. The aim of this paper is to define and study computationally efficient versions of CPS.

We start, in Section 2, from defining randomized predictive systems (RPS). In Section 3 we define their special case, split conformal predictive systems (SCPS), which are computationally efficient but may suffer loss of predictive efficiency as compared with CPS (which is indirectly confirmed in our experiments in Section 6, where SCPS suffer larger losses than their competitor that uses data more efficiently). An important advantage of SCPS is that they are, similarly to CPS, provably valid; a suitable notion of validity is defined in Section 3 and the validity of SCPS is demonstrated (by referring to a standard result).

In Section 4 we build cross-conformal predictive systems (CCPS) on top of split conformal predictive systems. In principle CCPS can lose their validity (and therefore, formally are no longer RPS), but in practice they usually satisfy the requirement of validity, as defined in Section 3 (cf. the experiments in Section 6 and Vovk 2015).

Section 6 is devoted to comparing the predictive efficiency of SCPS and CCPS and exploring the empirical validity of CCPS. In this paper, we measure predictive efficiency of predictive distributions using a loss function called continuous ranked probability score (CRPS). This loss function and the way it is applied in our context are defined in the preceding section, Section 5.

Section 7 concludes and gives a direction of further research.

2. Randomized predictive systems

Let $\mathbf{X}$ be an *object space*; we define the *observation space* as $\mathbf{Z} := \mathbf{X} \times \mathbb{R}$, each observation $z = (x, y) \in \mathbf{Z}$ consisting of an object $x \in \mathbf{X}$ and its label $y \in \mathbb{R}$.

We will use the following definition given in Vovk et al. (2017) (a modification of the definition in Shen et al. 2018, Definition 1). Let U be the uniform probability measure on the interval $[0, 1]$.

Definition 1 *A function $Q : \mathbf{Z}^{n+1} \times [0, 1] \to [0, 1]$ is called a* randomized predictive system *(RPS) if it satisfies the following three requirements:*

R1 *i For each training sequence $(z_1, \dots, z_n) \in \mathbf{Z}^n$ and each test object $x \in \mathbf{X}$, the function $Q(z_1, \dots, z_n, (x, y), \tau)$ is monotonically increasing both in y and in τ (where "monotonically increasing" is understood in the wide sense allowing intervals of constancy). In other words, for each $\tau \in [0, 1]$, the function*

$$y \in \mathbb{R} \mapsto Q(z_1, \dots, z_n, (x, y), \tau)$$

is monotonically increasing, and for each $y \in \mathbb{R}$, the function

$$\tau \in [0, 1] \mapsto Q(z_1, \dots, z_n, (x, y), \tau)$$

is also monotonically increasing.

ii For each training sequence $(z_1, \dots, z_n) \in \mathbf{Z}^n$ *and each test object* $x \in \mathbf{X}$,

$$\lim_{y \to -\infty} Q(z_1, \dots, z_n, (x, y), 0) = 0 \tag{1}$$

and

$$\lim_{y \to \infty} Q(z_1, \dots, z_n, (x, y), 1) = 1. \tag{2}$$

R2 For any probability measure P *on* $\mathbf{Z}$*, as function of random training observations* $z_1 \sim P, \dots, z_n \sim P$*, a random test observation* $z \sim P$*, and a random number* $\tau \sim U$*, all assumed independent, the distribution of* Q *is uniform:*

$$\forall \alpha \in [0, 1] : \mathbb{P}\{Q(z_1, \dots, z_n, z, \tau) \le \alpha\} = \alpha. \tag{3}$$

The output $y \in \mathbb{R} \mapsto Q(z_1, \dots, z_n, (x, y), \tau)$ of an RPS on a given training sequence $z_1, \dots, z_n$, test object x, and random number τ will be referred to as a *predictive distribution (function)*.

3. Split conformal predictive systems

In this section we will modify the definitions of conformal predictive systems given in Vovk et al. (2017) along the lines of Balasubramanian et al. (2014, Section 2.3) (removing an unnecessary assumption in Vovk et al. 2005, Section 4.1). A *split conformity measure* is a family of measurable function $A : \mathbf{Z}^{m+1} \to \mathbb{R} \cup \{-\infty, \infty\}$, $m = 1, 2, \dots$. The intention is that $A(z_1, \dots, z_{m+1})$ measures how large the label y_{m+1} in z_{m+1} is, as compared with the labels in $z_1, \dots, z_m$. Suppose the training sequence $z_1, \dots, z_n$ is split into two parts: the *proper training sequence* $z_1, \dots, z_m$ and the *calibration sequence* $z_{m+1}, \dots, z_n$; we are given a test object x. The output of the *split conformal transducer* determined by the split conformity measure A is defined as

$$\begin{aligned} Q(z_1, \dots, z_n, (x, y), \tau) := {} & \frac{1}{n - m + 1} \left| \{i = m + 1, \dots, n \mid \alpha_i < \alpha^y\} \right| \\ & + \frac{\tau}{n - m + 1} \left| \{i = m + 1, \dots, n \mid \alpha_i = \alpha^y\} \right| + \frac{\tau}{n - m + 1}, \end{aligned} \tag{4}$$

where the *conformity scores* α_i, $i = m + 1, \dots, n$, and α^y, $y \in \mathbb{R}$, are defined by

$$\begin{aligned} \alpha_i &:= A(z_1, \dots, z_m, (x_i, y_i)), \qquad i = m + 1, \dots, n, \\ \alpha^y &:= A(z_1, \dots, z_m, (x, y)). \end{aligned}$$

A function is a *split conformal transducer* if it is the split conformal transducer determined by some split conformity measure. A *split conformal predictive system* (SCPS) is a function which is both a split conformal transducer and a randomized predictive system.

The standard property of validity (satisfied automatically) for split conformal transducers is that the values $Q(z_1, \dots, z_n, z, \tau)$ are distributed uniformly on $[0, 1]$ when $z_1, \dots, z_n, z$ are IID and τ is generated independently of $z_1, \dots, z_n, z$ from the uniform probability distribution U on $[0, 1]$ (see, e.g., Vovk et al. 2005, Proposition 4.1).

It is much easier to get an RPS using split conformal transducers than using conformal transducers. A split conformity measure A is *isotone* if, for all m, $z_1, \dots, z_m$, and x, $A(z_1, \dots, z_m, (x, y))$ is isotone in y,

$$y \le y' \Longrightarrow A(z_1, \dots, z_m, (x, y)) \le A(z_1, \dots, z_m, (x, y')) \tag{5}$$

(cf. Vovk et al. 2017, the definition of monotonic conformity measures in Section 2). An isotone split conformity measure A is *balanced* if, for any m and $z_1, \dots, z_m$, the set

$$\operatorname{conv} A(z_1, \dots, z_m, (x, \mathbb{R})) := \operatorname{conv} \{A(z_1, \dots, z_m, (x, y)) \mid y \in \mathbb{R}\} \tag{6}$$

does not depend on x, where conv stands for the convex closure in $\mathbb{R}$. The set (6) then coincides with $\operatorname{conv} A(z_1, \dots, z_m, \mathbf{Z})$ and has one of four forms: (a, b), $[a, b)$, $(a, b]$, or $[a, b]$, where $a < b$ are elements of the extended real line $\mathbb{R} \cup \{-\infty, \infty\}$; typically, $\operatorname{conv} A(z_1, \dots, z_m, \mathbf{Z}) = (-\infty, \infty)$.

Proposition 2 *The split conformal transducer (4) based on a balanced isotone split conformity measure is an RPS.*

Proof Since property R2 is automatic, we only need to check R1. It is clear that (4) is increasing in τ (and linear).

To show that it is increasing in y, split, in the context of (4), all $i \in \{m+1, \dots, n\}$ into three groups: the i in group 1 satisfy $\alpha_i < \alpha^y$, the i in group 2 satisfy $\alpha_i = \alpha^y$, and the i in group 3 satisfy $\alpha_i > \alpha^y$. Then (4) is the total weight of all i where the weights are 1, $\tau \in [0, 1]$, and 0 for i in groups 1, 2, and 3, respectively. As y increases, α^y increases as well, and therefore, each i can only move to a lower-numbered group thus increasing (4).

Out of the remaining two conditions, let us check, e.g., (2). It suffices to notice that, since A is balanced, we have $\alpha^y \ge \max_{i \in \{m+1, \dots, n\}} \alpha_i$ from some y on, for any $z_1, \dots, z_n$ and x. ∎

The next proposition shows that a split conformity measure being isotone and balanced is not only a sufficient but also a necessary condition for the corresponding split conformal transducer to be an RPS.

Proposition 3 *If the split conformal transducer based on a split conformity measure A is an RPS, A is isotone and balanced.*

Proof Suppose A is not isotone. Fix m, $z_1, \dots, z_m$, x, y, and y' such that $y < y'$ but the consequent of (5) is violated. Then the putative predictive distribution $Q(z_1, \dots, z_m, (x, y), (x, \cdot), 1)$, corresponding to the proper training sequence $z_1, \dots, z_m$, calibration sequence (x, y), test object x, and $\tau = 1$, will not be increasing: its value at y (which is 1) will be greater than its value at y' (which is 0.5).

Now suppose A is not balanced. Fix m, $z_1, \dots, z_m$, and $x, x' \in \mathbf{X}$ such that

$$\operatorname{conv} A(z_1, \dots, z_m, (x, \mathbb{R})) \ne \operatorname{conv} A(z_1, \dots, z_m, (x', \mathbb{R}))$$

(cf. (6)). Suppose, for concreteness, that there is $y \in \mathbb{R}$ such that

$$\operatorname{conv} A(z_1, \dots, z_m, (x, \mathbb{R})) \ni y < \operatorname{conv} A(z_1, \dots, z_m, (x', \mathbb{R})),$$

Algorithm 1: Split Conformal Predictive System

Data: a training sequence $(x_i, y_i) \in \mathbf{Z}$, $i = 1, \dots, n$, and a test object $x \in \mathbf{X}$.
for $i \in \{1, \dots, n-m\}$ **do**
  define C_i by the condition $A(z_1, \dots, z_m, z_{m+i}) = A(z_1, \dots, z_m, (x, C_i))$
end
sort $C_1, \dots, C_{n-m}$ in the increasing order obtaining $C_{(1)} \le \dots \le C_{(n-m)}$;
set $C_{(0)} := -\infty$ and $C_{(n-m+1)} := \infty$;
return the predictive distribution (7) for the label y of x

where $y < S$ means $\forall s \in S : y < s$ when $S \subseteq \mathbb{R}$. (The other three possible cases can be analyzed in the same way.) Let $z_1, \dots, z_m$ be the proper training sequence, (x, y) be the calibration sequence, x' be the test object, and the random number be $\tau = 0$. Then we will have

$$\lim_{y' \to -\infty} Q(z_1, \dots, z_m, (x, y), (x', y'), 0) > 0,$$

which contradicts R1 (cf. (1)). ■

Let us say that a split conformity measure A is *strictly isotone* if (5) hold with both "$\le$" replaced by "$<$". A possible implementation of the SCPS based on a balanced strictly isotone split conformity measure is shown as Algorithm 1, where the predictive distribution is defined by

$$Q(z_1, \dots, z_n, (x, y), \tau) := \begin{cases} \frac{i+\tau}{n-m+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n-m\} \\ \frac{i'-1+(i''-i'+2)\tau}{n-m+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n-m\}, \end{cases} \tag{7}$$

where $i' := \min\{j \mid C_{(j)} = C_{(i)}\}$ and $i'' := \max\{j \mid C_{(j)} = C_{(i)}\}$. To use the terminology of Vovk et al. (2017), the thickness of this predictive distribution is $\frac{1}{n-m+1}$ with the exception size at most $n - m$.

How computationally efficient Algorithm 1 is depends on how easy to solve the equation defining C_i is. A standard choice is

$$A(z_1, \dots, z_m, (x, y)) := \frac{y - \hat{y}}{\hat{\sigma}}, \tag{8}$$

where $\hat{y}$ is a prediction for y computed from x and $z_1, \dots, z_m$ as training sequence, and $\hat{\sigma}$ is an estimate of the quality of $\hat{y}$ computed from the same data. In this case the equation

$$A(z_1, \dots, z_m, z_{m+i}) = A(z_1, \dots, z_m, (x, C_i)) \tag{9}$$

defining C_i becomes

$$\frac{y_{m+i} - \hat{y}_{m+i}}{\hat{\sigma}_{m+i}} = \frac{C_i - \hat{y}}{\hat{\sigma}},$$

where $\hat{y}_{m+1}$ (resp. $\hat{y}$) is the prediction for y_{m+1} (resp. y) computed from x_{m+1} (resp. x) and $z_1, \dots, z_m$ as training sequence, and $\hat{\sigma}_{m+1}$ (resp. $\hat{\sigma}$) is the estimate of the quality of $\hat{y}_{m+1}$ (resp. $\hat{y}$) computed from the same data. The last equation allows us to set

$$C_i := \hat{y} + \frac{\hat{\sigma}}{\hat{\sigma}_{m+i}} (y_{m+i} - \hat{y}_{m+i}) .$$

For more complicated split conformity measures A, it might be more efficient to use the expression (4) directly for a grid of values of y.

4. Cross-conformal predictive distributions

Remember that a *multiset* (or bag) is different from a set in that it can contain several copies of the same element. A split conformity measure A is a *cross-conformity measure* if $A(z_1, \dots, z_m, z)$ does not depend on the order of its first m arguments; in other words, if $A(z_1, \dots, z_m, z)$ only depends on the multiset $\lbag z_1, \dots, z_m \rbag$ and z (where $\lbag \cdots \rbag$ is used as the analogue of $\{\cdots\}$ for multisets).

Given a balanced isotone cross-conformity measure A, the corresponding *cross-conformal predictive system* (CCPS) is defined as follows. The training sequence $z_1, \dots, z_n$ is randomly split into K non-empty multisets (*folds*) z_{S_k}, $k = 1, \dots, K$, of equal (or as equal as possible) sizes, where $K \in \{2, 3, \dots\}$ is a parameter of the algorithm, $(S_1, \dots, S_K)$ is a partition of the index set $\{1, \dots, n\}$, and z_{S_k} consists of z_i, $i \in S_k$. For each $k \in \{1, \dots, K\}$ and each potential label $y \in \mathbb{R}$ of the test object x find the conformity scores of the observations in z_{S_k} and of (x, y) by

$$\alpha_{i,k} := A(z_{S_{-k}}, z_i), \quad i \in S_k, \qquad \alpha_k^y := A(z_{S_{-k}}, (x, y)),$$

where $S_{-k} := \cup_{j \neq k} S_j = \{1, \dots, n\} \setminus S_k$. The corresponding p-values and CCPS are defined by

$$p^y = Q(z_1, \dots, z_n, (x, y), \tau) := \frac{1}{n+1} \sum_{k=1}^{K} \left| \left\{ i \in S_k \mid \alpha_{i,k} < \alpha_k^y \right\} \right| + \frac{\tau}{n+1} \sum_{k=1}^{K} \left| \left\{ i \in S_k \mid \alpha_{i,k} = \alpha_k^y \right\} \right| + \frac{\tau}{n+1}. \quad (10)$$

The intuition behind (10) is that it becomes an SCPS when the training multisets $z_{S_{-k}}$ are replaced by a single hold-out training sequence (one disjoint from and independent of $z_1, \dots, z_n$).

An implementation of the CCPS based on a balanced strictly isotone cross-conformity measure is shown as Algorithm 2, where the predictive distribution is now defined by

$$Q(z_1, \dots, z_n, (x, y), \tau) := \begin{cases} \frac{i+\tau}{n+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n\} \\ \frac{i'-1+(i''-i'+2)\tau}{n+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n\}, \end{cases} \quad (11)$$

where, as before, $i' := \min\{j \mid C_{(j)} = C_{(i)}\}$ and $i'' := \max\{j \mid C_{(j)} = C_{(i)}\}$; the only difference from (7) is that we use n in place of $n - m$ (now all training observations are used

Algorithm 2: Cross-Conformal Predictive System

Data: a training sequence $(x_i, y_i) \in \mathbf{Z}$, $i = 1, \dots, n$, and a test object $x \in \mathbf{X}$.
split $z_1, \dots, z_n$ into K folds z_{S_k} as described in text;
set $C := \emptyset$, where C is a multiset;
for $k \in \{1, \dots, K\}$ **do**
  for $i \in S_k$ **do**
    define $C_{i,k}$ by the condition $A(z_{S_{-k}}, z_i) = A(z_{S_{-k}}, (x, C_{i,k}))$;
    put $C_{i,k}$ in C
  end
end
sort C in the increasing order obtaining $C_{(1)} \le \dots \le C_{(n)}$;
set $C_{(0)} := -\infty$ and $C_{(n+1)} := \infty$;
return the predictive distribution (11) for the label y of x

for calibration). The thickness of this predictive distribution is $\frac{1}{n+1}$ with the exception size at most n. The size of the multiset C in Algorithm 2 grows from 0 to n as the algorithm runs. As in the case of SCPS, it might be easier to use (10) directly if the equations defining $C_{i,k}$ are difficult to solve. (Alternatively, one could use (13) below instead of (10).)

Define a separate p-value

$$p_k^y := \frac{1}{|S_k| + 1} \left| \left\{ i \in S_k \mid \alpha_{i,k} < \alpha_k^y \right\} \right| + \frac{\tau}{|S_k| + 1} \left| \left\{ i \in S_k \mid \alpha_{i,k} = \alpha_k^y \right\} \right| + \frac{\tau}{|S_k| + 1} \tag{12}$$

for each fold; let us check that p^y is close to being an average of p_k^y. Comparing (10) and (12), we can see that

$$(n+1)p^y - \tau = \sum_{k=1}^{K} (|S_k| + 1)\, p_k^y - K\tau,$$

which implies

$$p^y = \sum_{k=1}^{K} \frac{|S_k| + 1}{n+1} p_k^y - \frac{K-1}{n+1}\tau. \tag{13}$$

The sum $\sum_{k=1}^{K} \dots$ is not quite a weighted average since the sum of the weights is slightly above 1 ("slightly" assumes $K \ll n$), but this is partially compensated by the subtrahend in (13); overall, the right-hand side of (13) is a weighted average of p_k^y and τ, with the weight in front of τ being negative.

According to the intuition behind cross-conformal predictive distributions described earlier, we will get perfect validity for CCPS if we replace the K training multisets (the complements to the K folds) by one hold-out training sequence. But whereas SCPS are provably valid, in the sense of being RPS, real CCPS are not RPS: see the example in Vovk (2015, Appendix A). In experimental studies, this phenomenon has been demonstrated by Linusson et al. (2017), who showed the danger of randomized and extremely unstable underlying algorithms. (Perhaps such unstable algorithms might be stabilized, to some

degree, by using the same seed of the random numbers generator for each fold, or by averaging conformity scores over several seeds, or both.) A useful intuition (Linusson et al., 2017) is that the random p-values coming from different folds (and then essentially averaged by conformal predictors) are to some degree independent, and so the distribution of cross-conformal p-values is intermediate between the uniform and the Bates distributions; therefore, cross-conformal p-values are conservative when not exact (for small significance levels). According to a result in Vovk and Wang (2018, Corollary 2 with $r := 1$), we will get provably valid (but perhaps conservative) p-values if we multiply the p-values output by a cross-conformal transducer by 2; the empirical fact observed by Linusson et al. (2017) is that for randomized and unstable underlying algorithms even unadjusted p-values output by a cross-conformal transducer are valid but perhaps overly conservative for interesting (not exceeding 0.5) significance levels.

A more general procedure than the cross-conformal predictor was proposed in Carlsson et al. (2014) under the name of "aggregated conformal predictor". Similar methods might be applicable for producing conformal predictive distributions.

5. Continuous ranked probability score

Suppose the prediction for a label $y \in \mathbb{R}$ is a distribution function $F : \mathbb{R} \to [0,1]$ and the observed value of y is y_i. The quality of the prediction F in view of the actual outcome y_i is often measured by the *continuous ranked probability score*

$$\mathrm{CRPS}(F, y_i) := \int_{-\infty}^{\infty} \left(F(y) - \mathbf{1}_{\{y \geq y_i\}}\right)^2 \mathrm{d}y, \tag{14}$$

where $\mathbf{1}$ stands for the indicator function. The lowest possible value 0 is attained when F is concentrated at y_i, and in all other cases $\mathrm{CRPS}(F, y_i)$ will be positive. (See, e.g., Gneiting and Katzfuss 2014 for further details and references.)

Strictly speaking, (14) is not applicable to split and cross-conformal predictive distributions, which are somewhat "fuzzy" (the thickness for the former is $\frac{1}{n-m+1}$ and for the latter it is $\frac{1}{n+1}$). Therefore, instead of (7) and (11) we use their crisp modifications

$$Q(z_1, \dots, z_n, (x, y)) := \begin{cases} \frac{i}{n-m} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n-m\} \\ \frac{i}{n-m} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n-m\} \end{cases} \tag{15}$$

and

$$Q(z_1, \dots, z_n, (x, y)) := \begin{cases} \frac{i}{n} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n\} \\ \frac{i}{n} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n\}, \end{cases} \tag{16}$$

respectively; these modifications no longer depend on τ. In cases where the equation (9) or its analogue for the CCPS are difficult to solve, we could instead use the following crisp modifications of (4) and (10), respectively:

$$Q(z_1, \dots, z_n, (x, y)) := \frac{1}{n-m} \left|\{i = m+1, \dots, n \mid \alpha_i \leq \alpha^y\}\right|,$$

$$Q(z_1,\dots,z_n,(x,y)) := \frac{1}{n}\sum_{k=1}^{K}\left|\left\{i \in S_k \mid \alpha_{i,k} \le \alpha_k^y\right\}\right|.$$

The last equation, defining a crisp CCPS, can be rewritten as

$$Q(z_1,\dots,z_n,(x,y)) = \sum_{k=1}^{K}\frac{|S_k|}{n}p_k^y$$

(cf. (13)), where the separate "p-values" for each fold are now defined as

$$p_k^y := \frac{1}{|S_k|}\left|\left\{i \in S_k \mid \alpha_{i,k} \le \alpha_k^y\right\}\right|$$

(they, however, do not satisfy any validity properties).

6. Experiments

Given a training sequence $(z_1,\dots,z_n)$ and a test sequence $(z_{n+1},\dots,z_{n+k})$, the quality of prediction is represented by the distribution of $\mathrm{CRPS}(F_i,y_i)$, $i=n+1,\dots,n+k$, where F_i is the predictive distribution for the label y_i of the test object x_i. The length k of the test sequence will be 100 in all our experiments. In order to obtain boxplots less affected by the random split of the training sequence into proper training and calibration sequences (in the case of SCPS) or K folds (in the case of CCPS), for each boxplot we perform 20 random splits and for each split find 100 values $\mathrm{CRPS}(F_i,y_i)$ for all test observations (the same test sequence is used for each split); the resulting boxplot is based on all 2000 numbers (except for the combination of CCPS, which are less affected by randomness, and Random Forest, which is more computationally demanding, in which case we perform 10 random splits).

In our experiments we use two well-known benchmark data sets, namely `Boston Housing` and `Diabetes` available at http://scikit-learn.org/stable/datasets/.

The `Boston Housing` data set consists of 506 observations each with 14 attributes. We randomly split the full data set into a training sequence of length $n=406$ and a test sequence of length 100. The first two figures, Figures 1–2, use Least Squares linear regression as the underlying algorithm. They show the performance of the SCPS and CCPS determined by the cross-conformity measure (a special case of (8))

$$A(z_1,\dots,z_m,(x,y)) := y - \hat{y}, \tag{17}$$

where $\hat{y}$ is the Least Squares prediction for the label of x based on $z_1,\dots,z_m$ as training sequence. (Remember that each cross-conformity measure is also a split conformity measure.)

For Figure 1, we randomly split the training sequence into proper training and calibration sequences of equal lengths and plot the split conformal predictive distribution for the first test object. We show it separately for $\tau=0$ and $\tau=1$: the former is represented as a blue solid line and the latter as a red dashed line. As expected, the two lines are close to each other: we have already mentioned that the thickness of the split conformal predictive distribution is $\frac{1}{n-m+1}$, and in this case it evaluates to $\frac{1}{406-203+1}\approx 0.005$. This example is

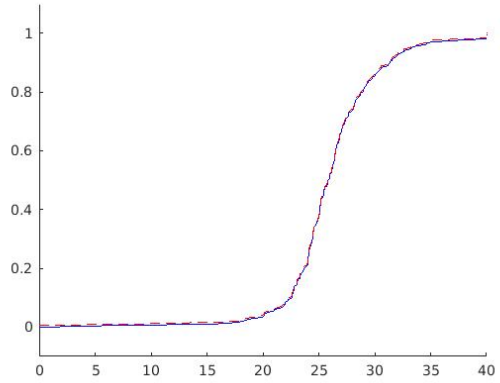

Figure 1: The split conformal predictive distribution for the first test object of the `Boston Housing` data set, the Least Squares underlying algorithm, and a 50%:50% split of the training sequence into proper training and calibration sequences, as described in text. The blue solid line corresponds to $\tau = 0$ and the red dashed line to $\tau = 1$.

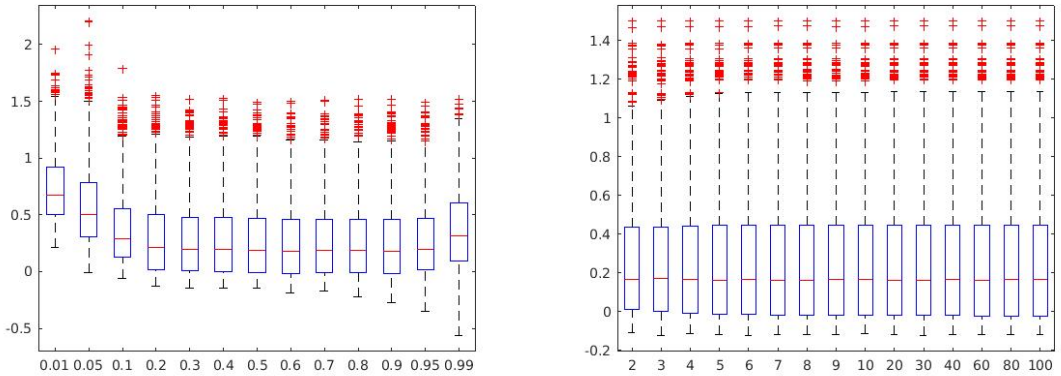

Figure 2: The performance of the SCPS (left panel) and CCPS (right panel) on the `Boston Housing` data set using Least Squares as the underlying algorithm, as described in text. The vertical axis gives the CRPS on the base-10 log scale. Left panel: the numbers on the horizontal axis are the fractions m/n of the training sequence used as the proper training sequence. Right panel: the numbers on the horizontal axis are the numbers K of folds.

typical, and it is the only empirical conformal predictive distribution that we show in this paper; all other figures and tables will show aggregated results.

To produce the left panel of Figure 2, we randomly split the training sequence into a proper training sequence of length m and a calibration sequence of length $n - m$ for a

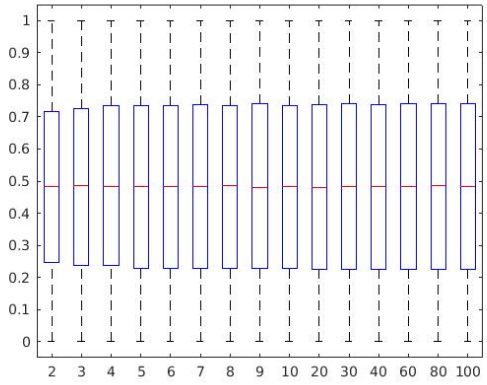

Figure 3: The distribution of $Q(z_1, \ldots, z_n, (x, y))$ over the test set for the CCPS on the `Boston Housing` data set using Least Squares as the underlying algorithm. The numbers on the horizontal axis are the numbers K of folds.

range of values of the fraction m/n. The horizontal axis is labelled by fraction m/n, which is in fact the parameter of the algorithm (for a given value of m/n we find m and then round it to the nearest integer). For each value of m/n we perform 20 random splits, for each split compute the CRPS loss of the (crisp) SCPS based on (17) and Least Squares on each observation in the test sequence and represent the resulting 2000 CRPS losses as a boxplot. The values of m/n used in our experiments are between 0.1 and 0.9, plus a few more extreme values. We can see a characteristic U-shape (especially pronounced on the left), with 50%:50% splits giving reasonable results. The range of values of m/n giving similar results is quite wide. Notice that large (close to 1) values of m/n not only lead to a large CRPS loss (not as large as for small values of m/n) but also make the CRPS loss less stable.

The right panel of Figure 2 is similar to the left panel, but now we use the CCPS and label the horizontal axis by the number K of folds. The usual advice in cross validation is to use $K \in \{5, 10\}$, and these two values produce reasonable results. In fact, the results are remarkably stable and barely depend on K.

A natural question is whether the CCPS satisfy the property of validity R2 at least approximately; remember that there are no theoretical validity results for cross-conformal predictors, and it has been demonstrated theoretically (Vovk, 2015, Appendix A) and experimentally (Linusson et al., 2017) that a loss of validity is possible. Figure 3 gives the distribution of the values (16), where $z_1, \ldots, z_n$ is the training set, and (x, y) range over the test set. The lower and upper quartiles of the distributions are approximately 0.25 and 0.75 for all numbers K of folds, which is consistent with R2 (stipulating the uniform distributions). Figure 4 gives fuller information for $K = 5$ and $K = 100$ (the former being one of the two standard values, and the latter being the maximal value given in Figure 3). Namely, it gives the *calibration curves*, which are the sets of points $(\alpha, F(\alpha))$, $\alpha \in (0, 1)$ ranging over the possible significance levels and $F(\alpha)$ being the percentage of the values

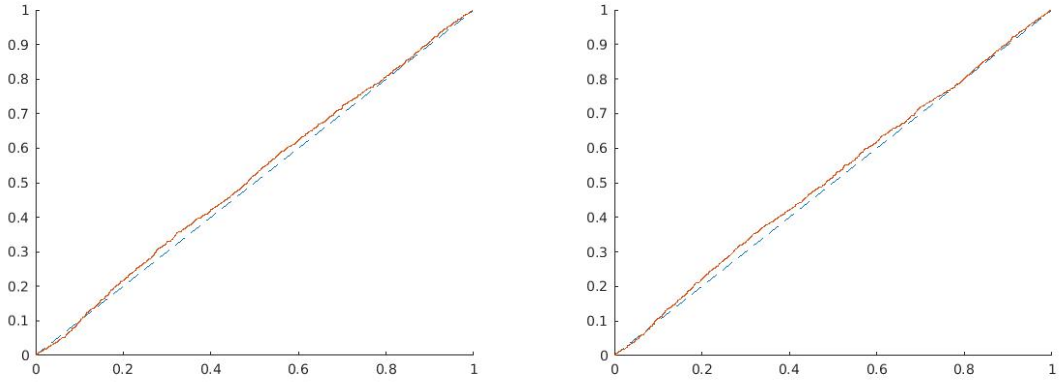

Figure 4: The calibration curves (described in text) for the CCPS on the `Boston Housing` data set using Least Squares as the underlying algorithm. On the left panel, $K = 5$, and on the right panel, $K = 100$.

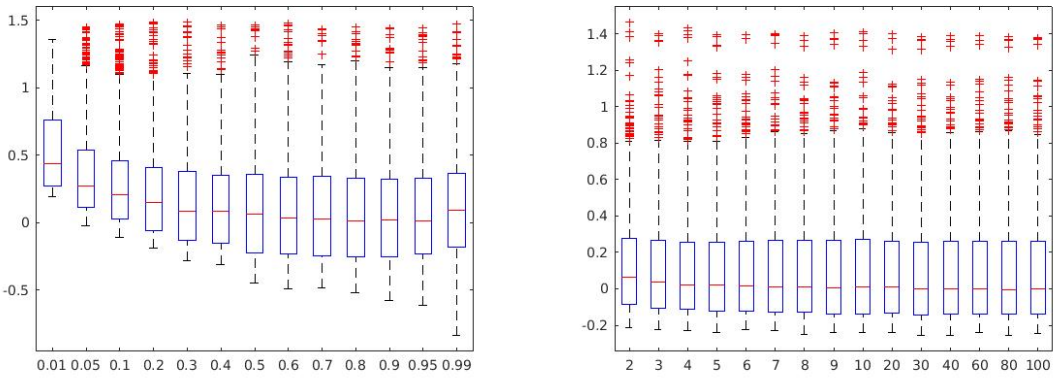

Figure 5: The analogue of Figure 2 for the `Boston Housing` data set and the Random Forest underlying algorithm.

$Q(z_1, \dots, z_n, (x, y))$ for (x, y) in the test set that do not exceed α. Under perfect validity (3) and an infinitely large test set, the calibration curves should be the diagonals shown as dashed lines on both panels of Figure 3; the actual calibration curves are fairly close. The calibration curve for $K = 10$ is very similar.

Figure 5 is the analogue of Figure 2 in which the Least Squares method is replaced by Random Forest (TreeBagger in MATLAB, with the number of trees set to 20, which is a standard value given in the instructions); we are still using the cross-conformity measure (17). We can see that non-linear machine learning algorithms used as underlying algorithms

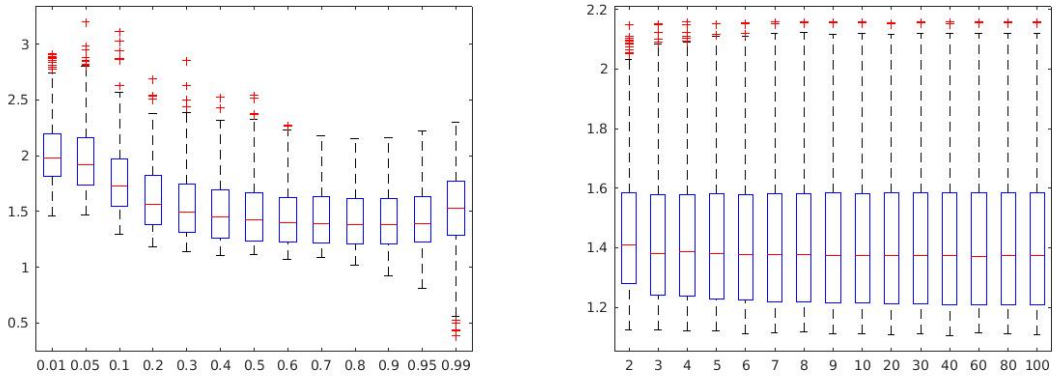

Figure 6: The analogue of Figure 2 for the `Diabetes` data set and Least Squares as the underlying algorithm.

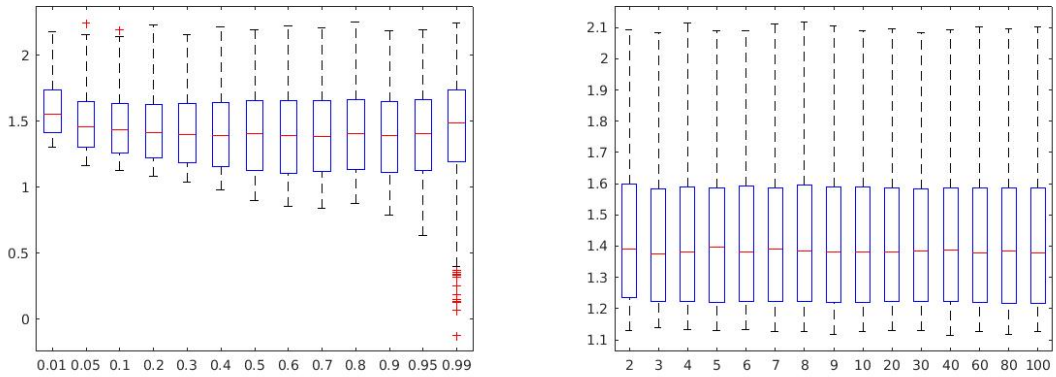

Figure 7: The analogue of Figure 2 for the `Diabetes` data set and Random Forest as the underlying algorithm.

can improve performance as measured by CRPS. The left panel of Figure 5 shows that 50%:50% splits remain competitive, and the right panel shows that $K \in \{5, 10\}$ are good choices for the number of folds.

The `Diabetes` data set consists of 10 physiological measures on 442 patients, and the label indicates disease progression after one year. We split randomly the whole data set into a training sequence of length 342 and a test sequence of length 100. Figures 6 and 7 are the analogues of Figures 2 and 5 for this data set and this split into training and test sequences. They confirm that 50%:50% splits of the training sequence produces reasonable results for SCPS and that $K \in \{5, 10\}$ are reasonable numbers of folds for CCPS.

Table 1: Best results for the median CRPS loss for SCPS and CCPS for the two data sets and two underlying algorithms.

Data set, underlying algorithm	SCPS	CCPS
Boston Housing, Least Squares	1.5010	1.4526
Boston Housing, Random Forest	1.0255	0.9892
Diabetes, Least Squares	24.0280	23.5547
Diabetes, Random Forest	24.2302	23.7011

The best results presented in Figures 2 and 5–7 are summarized in Table 1. Namely, the table reports the median CRPS losses shown in Figures 2 and 5–7 obtained by optimizing the parameters m/n in the case of SCPS and K in the case of CCPS. As discussed earlier, both SCPS and CCPS are fairly insensitive to choosing their parameters, and so the best results given in Table 1 are in fact typical. We can see that whereas using the non-linear method is beneficial in the case of `Boston Housing`, it is not in the case of `Diabetes`. In all cases CCPS perform better than SCPS.

Not only is the efficiency of the CCPS with respect to the CRPS loss better than that of the SCPS, it can also be argued that the CCPS may be safer from the point of view of validity. Suppose that, for some reason, we would like to avoid randomization and use (15) (in the case of SCPS) or (16) (in the case of CCPS) instead of (7) or (11), respectively. We saw that the CCPS is still empirically valid in our experiments, even in the extreme case of $K = 100$. Figure 3 is for the `Boston Housing` data set with Least Squares as the underlying algorithm, but we get very similar plots when we replace the `Boston Housing` data set by `Diabetes`, when we replace Least Squares by Random Forest, and when we do both. On the other hand, when using (15) in place of (7), the SCPS lose not only theoretical but also empirical validity. For example, for `Boston Housing` and $m/n = 0.99$ (the right end of the horizontal axis in the left panel of Figure 2), the size of the calibration set is 4, and so the empirical predictive distribution (15) only takes values in $\{0, 0.25, 0.5, 0.75, 1\}$; the distribution of its values at the true labels is clearly very different from being uniform.

7. Conclusion

In this paper we have given definitions and described ways of computing split and cross-conformal predictive distributions. We have studied their empirical performance using two small benchmark data sets. Cross-conformal predictive distributions are more efficient and, in their non-randomized version, sometimes closer to being valid. It would be interesting to check the validity of our conclusions on a wider range of data sets.

Acknowledgments

We are very grateful to the three anonymous referees of the conference version of this paper for their thoughtful comments. This work has been supported by the EU Horizon

2020 Research and Innovation programme (grant 671555), AstraZeneca grant "Machine Learning for Chemical Synthesis" (R10911), Leverhulme Magna Carta Doctoral Centre, and Technology Integrated Health Management project awarded to the School of Mathematics and Information Security at Royal Holloway University of London as part of an initiative by NHS England supported by Innovate UK.

References

Vineeth N. Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk, editors. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations, and Applications*. Elsevier, Amsterdam, 2014.

Lars Carlsson, Martin Eklund, and Ulf Norinder. Aggregated conformal prediction. In Lazaros Iliadis, Ilias Maglogiannis, Harris Papadopoulos, Spyros Sioutas, and Christos Makris, editors, *AIAI Workshops, COPA 2014*, volume 437 of *IFIP Advances in Information and Communication Technology*, pages 231–240, Berlin, 2014. Springer.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

Henrik Linusson, Ulf Norinder, Henrik Boström, Ulf Johansson, and Tuve Löfström. On the calibration of aggregated conformal predictors. *Proceedings of Machine Learning Research*, 60:154–173, 2017. COPA 2017.

Ilia Nouretdinov, Denis Volkhonskiy, Pitt Lim, Paolo Toccaceli, and Alexander Gammerman. Inductive Venn-Abers predictive distribution. *Proceedings of Machine Learning Research*, 60:15–36, 2018. COPA 2018.

Tore Schweder and Nils L. Hjort. *Confidence, Likelihood, Probability: Statistical Inference with Confidence Distributions*. Cambridge University Press, Cambridge, UK, 2016.

Jieli Shen, Regina Liu, and Minge Xie. Prediction with confidence—a general framework for predictive inference. *Journal of Statistical Planning and Inference*, 195:126–140, 2018.

Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.

Vladimir Vovk and Claus Bendtsen. Conformal predictive decision making. *Proceedings of Machine Learning Research*, 91:52–62, 2018. COPA 2018.

Vladimir Vovk and Ruodu Wang. Combining p-values via averaging. Technical Report arXiv:1212.4966 [math.ST], arXiv.org e-Print archive, April 2018.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Minge Xie. Nonparametric predictive distributions based on conformal prediction. *Proceedings of Machine Learning Research*, 60:82–102, 2017. COPA 2017.