Computationally efficient versions of conformal predictive distributions

Vladimir Vovk, Ivan Petej, Ilia Nouretdinov,
Valery Manokhin, and Alex Gammerman

практические выводы
теории вероятностей
могут быть обоснованы
в качестве следствий
гипотез о *предельной*
при данных ограничениях
сложности изучаемых явлений

On-line Compression Modelling Project (New Series)

Working Paper #22

First posted March 30, 2018. Last revised November 3, 2019.

Project web site:
http://alrw.net

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. The main advantage of split conformal predictive systems is their guaranteed validity, whereas for cross-conformal predictive systems validity only holds empirically and in the absence of excessive randomization. The main advantage of cross-conformal predictive systems is their greater predictive efficiency.

Contents

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 are only applicable to classification problems, 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 use for both singular and plural) in [21], based on the parallel work on predictive distributions in parametric statistics (see, e.g., [10, Chapter 12] and [11]). 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 essentially becoming probabilities. This facilitates new uses of conformal prediction, such as automatic decision making [14]. However, for many underlying algorithms CPS (like conformal predictors in general) are computationally inefficient: CPS require re-training the underlying algorithm for each test object and each postulated label for it, and this can be done efficiently only for a narrow class of underlying algorithms, including Least Squares [21] and Kernel Ridge Regression [17]. The main aim of this paper is to define and study computationally efficient versions of CPS without restrictions on the underlying algorithm.

A very recent development in Venn prediction has been the introduction, in the terminology of this paper, of split Venn–Abers predictive systems in [8] (COPA 2018), which are another way to produce predictive distributions. A secondary aim of this paper is to explore several versions of Venn–Abers predictive systems and compare them with conformal predictive systems.

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 typically 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; in Section 3, a suitable notion of validity is defined 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 [13] and Section 6).

Section 6 is devoted to comparing the predictive efficiency of SCPS and

CCPS and exploring their empirical validity. 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.

Sections 7 and 8 discuss more general issues (to be described momentarily). Section 9 concludes and gives directions of further research.

The conference version of this paper was published in the Proceedings of COPA 2018 [16], and the journal version is to be published in *Neurocomputing*. As compared with the conference version, in the journal version we added a more detailed comparison of SCPS and CCPS, a detailed discussion of Venn–Abers predictive systems (in Section 8), and the analysis of universality of various predictive systems (in Section 7). An important finding here is that SCPS and CCPS are universal, whereas Venn–Abers predictive systems are not.

2 Randomized predictive systems

Fix a nonempty measurable space $\mathbf{X}$; we will refer to it as our *object space*. Define the *observation space* as $\mathbf{Z} := \mathbf{X} \times \mathbb{R}$; each observation $z = (x, y) \in \mathbf{Z}$ consists of an object $x \in \mathbf{X}$ and its label $y \in \mathbb{R}$.

We will use the following definition, given in [21] (a modification of the definition in [11, 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 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

$$y \in \mathbb{R} \mapsto Q(z_1, \dots, z_n, (x, y), \tau) \tag{1}$$

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 $x \in \mathbf{X}$,

$$\lim_{y \to -\infty} Q(z_1, \dots, z_n, (x, y), 0) = 0 \tag{2}$$

and

$$\lim_{y \to \infty} Q(z_1, \dots, z_n, (x, y), 1) = 1. \tag{3}$$

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

$$\forall \alpha \in [0,1] : \mathbb{P}\left(Q(z_1, \ldots, z_n, z, \tau) \leq \alpha\right) = \alpha. \tag{4}$$

The output

$$y \in \mathbb{R} \mapsto Q(z_1, \ldots, z_n, (x,y), \tau) \tag{5}$$

of an RPS on a given training sequence $z_1, \ldots, z_n$, 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 [21] along the lines of [1, Section 2.3] (removing an unnecessary assumption in [15, Section 4.1]). A *split conformity measure* is a family of measurable functions $A_m : \mathbf{Z}^{m+1} \to \mathbb{R} \cup \{-\infty, \infty\}$, $m = 1, 2, \ldots$. The intention is that $A_m(z_1, \ldots, z_{m+1})$ measures how large the label y_{m+1} in z_{m+1} is, as compared with the labels in $z_1, \ldots, z_m$. Suppose the training sequence $z_1, \ldots, z_n$ is split into two parts: the *training sequence proper* $z_1, \ldots, z_m$ and the *calibration sequence* $z_{m+1}, \ldots, z_n$; we are given a test object x. The output of the *split conformal transducer* determined by the split conformity measure A is defined as

$$\begin{aligned} Q(z_1, \ldots, z_n, (x,y), \tau) := {} & \frac{1}{n-m+1} \left|\{i = m+1, \ldots, n \mid \alpha_i < \alpha^y\}\right| \\ & + \frac{\tau}{n-m+1} \left|\{i = m+1, \ldots, n \mid \alpha_i = \alpha^y\}\right| + \frac{\tau}{n-m+1}, \end{aligned} \tag{6}$$

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

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

(We omit the lower index m in A_m since it is determined by the number of arguments.) 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 $Q(z_1, \ldots, z_n, z, \tau)$ are distributed uniformly on $[0,1]$ when $z_1, \ldots, z_n, z$ are IID and τ is generated independently of $z_1, \ldots, z_n, z$ from the uniform probability distribution U on $[0,1]$ (see, e.g., [15, 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 *isotonic* if, for all m, $z_1, \dots, z_m$, and x, $A(z_1, \dots, z_m, (x, y))$ is isotonic in y, i.e.,

$$y \le y' \Longrightarrow A(z_1, \dots, z_m, (x, y)) \le A(z_1, \dots, z_m, (x, y')) \tag{7}$$

(cf. [21], the definition of monotonic conformity measures in Section 2). An isotonic split conformity measure A is *balanced* if, for any m and $z_1, \dots, z_m$, the set

$$\operatorname{conv} A(z_1, \dots, z_m, (x, \mathbb{R})) := \operatorname{conv} \{A(z_1, \dots, z_m, (x, y)) \mid y \in \mathbb{R}\} \tag{8}$$

does not depend on x, where conv stands for the convex closure in $\mathbb{R}$. The set (8) 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 $\mathbb{R} \cup \{-\infty, \infty\}$; in this paper, we will be mainly interested in the case $\operatorname{conv} A(z_1, \dots, z_m, \mathbf{Z}) = (-\infty, \infty)$.

Proposition 1. *The split conformal transducer* (6) *based on a balanced isotonic split conformity measure is an RPS.*

Proof. Since property R2 is automatic, we only need to check R1. It is clear that (6) is increasing in τ (and linear).

To show that it is increasing in y, split, in the context of (6), 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 (6) 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 (6).

Out of the remaining two conditions, let us check, e.g., (3). 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 isotonic and balanced is not only a sufficient but also a necessary condition for the corresponding split conformal transducer to be an RPS.

Proposition 2. *If the split conformal transducer based on a split conformity measure A is an RPS, A is isotonic and balanced.*

Proof. Suppose A is not isotonic. Fix m, $z_1, \dots, z_m$, x, y, and y' such that $y < y'$ but the consequent of (7) is violated. Then the putative predictive distribution $Q(z_1, \dots, z_m, (x, y), (x, \cdot), 1)$, corresponding to the training sequence proper $z_1, \dots, z_m$, 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

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

Algorithm 1 Split Conformal Predictive System

```
Require: A training sequence $(x_i, y_i) \in \mathbf{Z}$, $i = 1, \dots, n$.
Require: 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 for
  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 (9) for the label $y$ of $x$.
```

(cf. (8)). Suppose, for concreteness, that there is $y \in \mathbb{R}$ such that

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

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 the training sequence proper be $z_1, \dots, z_m$, the calibration sequence be (x, y), the test object be x', 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. (2)). □

Let us say that a split conformity measure A is *strictly isotonic* if (7) holds with both "$\le$" replaced by "$<$". A possible implementation of the SCPS based on a balanced strictly isotonic split conformity measure is shown as Algorithm 1, where the predictive distribution is defined by

$$Q(z_1, \dots, z_n, (x, y), \tau) := \begin{cases} \frac{i+\tau}{n-m+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n-m\} \\ \frac{i'-1+(i''-i'+2)\tau}{n-m+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n-m\}, \end{cases} \tag{9}$$

where $i' := \min\{j \mid C_{(j)} = C_{(i)}\}$ and $i'' := \max\{j \mid C_{(j)} = C_{(i)}\}$. To use the terminology of [21], 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 of split conformity measure is

$$A(z_1, \dots, z_m, (x, y)) := \frac{y - \hat{y}}{\hat{\sigma}}, \tag{10}$$

where $\hat{y}$ is a prediction for the label y computed from x as test object 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

$$A(z_1, \dots, z_m, z_{m+i}) = A(z_1, \dots, z_m, (x, C_i)) \tag{11}$$

defining C_i becomes

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

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

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

For more complicated split conformity measures A, it might be more efficient to use the expression (6) 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 $\wr z_1,\dots,z_m \int$ and z (where $\wr \cdots \int$ is used as the analogue of $\{\cdots\}$ for multisets).

Given a balanced isotonic 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 all 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 $S_{-k} := \cup_{j\ne k} S_j = \{1,\dots,n\}\setminus S_k$. The corresponding p-values and CCPS are defined by

$$\begin{aligned} p^y = Q(z_1,\dots,z_n,(x,y),\tau) := {} & \frac{1}{n+1}\sum_{k=1}^K \left|\{i\in S_k \mid \alpha_{i,k} < \alpha_k^y\}\right| \\ & + \frac{\tau}{n+1}\sum_{k=1}^K \left|\{i\in S_k \mid \alpha_{i,k} = \alpha_k^y\}\right| + \frac{\tau}{n+1}. \end{aligned} \tag{12}$$

The intuition behind (12) 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 $z_1,\dots,z_n$).

Algorithm 2 Cross-Conformal Predictive System

```
Require: A training sequence $(x_i, y_i) \in \mathbf{Z}$, $i = 1, \dots, n$.
Require: 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 for
  end for
  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 (13) for the label $y$ of $x$.
```

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

$$Q(z_1, \dots, z_n, (x, y), \tau) := \\ \begin{cases} \frac{i+\tau}{n+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \dots, n\} \\ \frac{i'-1+(i''-i'+2)\tau}{n+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \dots, n\}, \end{cases} \tag{13}$$

where, as before, $i' := \min\{j \mid C_{(j)} = C_{(i)}\}$ and $i'' := \max\{j \mid C_{(j)} = C_{(i)}\}$; the only difference from (9) is that we use n in place of $n - m$ (now all training observations are used 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 (12) directly if the equations defining $C_{i,k}$ are difficult to solve. (Alternatively, one could use (15) below instead of (12).)

Define a separate p-value

$$p_k^y := \frac{1}{|S_k| + 1} \left|\{i \in S_k \mid \alpha_{i,k} < \alpha_k^y\}\right| \\ + \frac{\tau}{|S_k| + 1} \left|\{i \in S_k \mid \alpha_{i,k} = \alpha_k^y\}\right| + \frac{\tau}{|S_k| + 1} \tag{14}$$

for each fold (cf. (6)); let us check that p^y is close to being an average of p_k^y. Comparing (12) and (14), we can see that

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

which implies

$$p^y = \sum_{k=1}^{K} \frac{|S_k|+1}{n+1} p_k^y - \frac{K-1}{n+1}\tau. \tag{15}$$

The sum $\sum_{k=1}^{K} \dots$ is not quite a weighted average of p_k^y since the sum of the weights is slightly above 1 ("slightly" assumes $K \ll n$), but this is partially compensated by the subtrahend in (15); overall, the right-hand side of (15) 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 [13, Appendix A]. In experimental studies, this phenomenon has been demonstrated by [6], 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 [6] is that the random p-values coming from different folds (and then essentially averaged by cross-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 [22] (see, e.g., Table 1 for $r := 1$), 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 [6] 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 [3] 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*

$$\mathrm{CRPS}(F, y_i) := \int_{-\infty}^{\infty} \left(F(y) - \mathbf{1}_{\{y \ge y_i\}}\right)^2 \, \mathrm{d}y, \tag{16}$$

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., [5] for further details and references.)

Strictly speaking, (16) is not applicable to split and cross-conformal predictive distributions, which are somewhat "fuzzy" (the thickness for the former

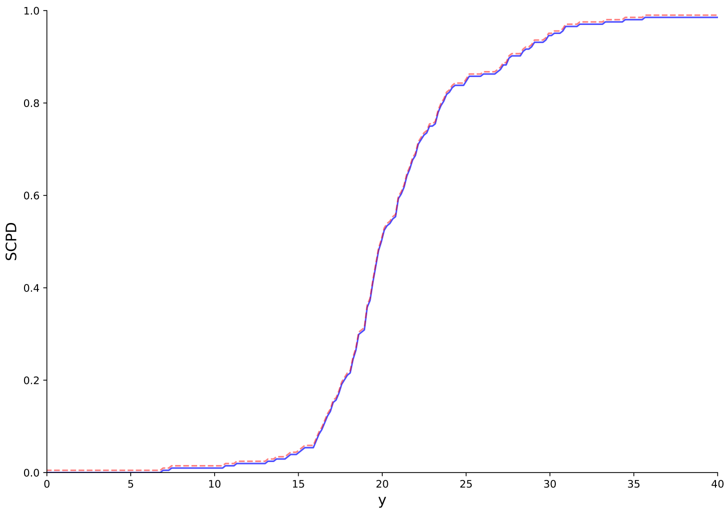

Figure 1: The split conformal predictive distribution for a random test object in the `Boston Housing` dataset (described in Section 6), the Least Squares underlying algorithm, and a random 50% : 50% split of the training sequence into proper training and calibration sequences. The blue solid line corresponds to $\tau = 0$ and the red dashed line to $\tau = 1$.

is $\frac{1}{n-m+1}$ and for the latter it is $\frac{1}{n+1}$). In practice, the fuzziness can usually be ignored, even for relatively small datasets: see, e.g., Figure 1. However, conceptually we do need to change split and cross-conformal predictive distributions slightly to remove their fuzziness.

Instead of (9) and (13) we use their crisp modifications

$$Q(z_1, \ldots, z_n, (x, y)) := \begin{cases} \frac{i}{n-m} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \ldots, n-m\} \\ \frac{i}{n-m} & \text{if } y = C_{(i)} \text{ and } y \neq C_{(i+1)} \text{ for } i \in \{1, \ldots, n-m\} \end{cases} \tag{17}$$

and

$$Q(z_1, \ldots, z_n, (x, y)) := \begin{cases} \frac{i}{n} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, \ldots, n\} \\ \frac{i}{n} & \text{if } y = C_{(i)} \text{ and } y \neq C_{(i+1)} \text{ for } i \in \{1, \ldots, n\}, \end{cases} \tag{18}$$

respectively; these modifications no longer depend on τ, and the convention for $y = C_{(i)}$ does not affect the value of CRPS. In cases where the equation (11) or its analogue for the CCPS are difficult to solve, we can instead use the following crisp modifications of (6) and (12), respectively:

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

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

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

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

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

6 Experiments

The purpose of this section is to compare the predictive performance of SCPS and CCPS and to recommend the choice of the parameter K for CCPS. Our choice of conformity measures might well be improved in future work (cf. Section 7).

In our experiments we use five well-known benchmark datasets, namely `Boston Housing`, `Diabetes`, `Yacht Hydrodynamics`, `Wine Quality`, and `Condition Based Maintenance of Naval Propulsion Plants` (abbreviated to `Naval Propulsion`) available at http://scikit-learn.org/stable/datasets/ (the first two) and the UCI Machine Learning repository [4] (the other ones). The first three datasets are small: `Boston Housing` consists of 506 observations, `Diabetes` of 442 observations, and `Yacht Hydrodynamics` of 308 observations; for them we use test sequences of length $l := 100$. The `Wine Quality` dataset consists of 6497 observations, and we use test sequences of length $l := 1000$. Finally, the `Naval Propulsion` dataset consists of 11,934 observations, and we use test sequences of length $l := 4000$.

Given a training sequence $(z_1,\dots,z_n)$ (where $n \in \{406, 342, 208, 5497, 7934\}$) and a test sequence $(z_{n+1},\dots,z_{n+l})$, the quality of prediction is represented by the distribution of $\text{CRPS}(F_i, y_i)$, $i = n+1,\dots,n+l$, where F_i is the predictive distribution for the label y_i of the test object x_i. As already mentioned, the length l of the test sequence is 100, 1000, or 4000 in our experiments.

In order to obtain boxplots less affected by the split of each dataset into a training and test sequence and by the random split of each training sequence into a training sequence proper and a calibration sequence (in the case of SCPS) or K folds (in the case of CCPS), we use the procedure given as Algorithm 3. Each dataset is randomly permuted 10 times. The last l observations of each permutation are used for testing and the rest for training. The first m observations in the training sequence are used as training sequence proper in the case of SCPS and consecutive blocks of the training sequence are used as the

K folds in the case of CCPS (using the `scikit-learn KFold` procedure with no randomization). The boxplots in all figures given below are indexed by the fractions m/n of the training sequence used as the training sequence proper (in the case of SCPS) or by the numbers K of folds (in the case of CCPS). For each split and each boxplot we find the l values $\text{CRPS}(F_i, y_i)$ for all test observations (the same test sequence is used for each split); the resulting boxplot is based on all $10\,l$ numbers.

The loops in lines 1, 6, 20, 29, 36, and 41 of Algorithm 3 are over our underlying algorithms U (Least Squares, Random Forest, and Neural Networks, as implemented in `scikit-learn`). In all cases the SCPS and CCPS use the cross-conformity measure (a special case of (10))

$$A(z_1, \dots, z_m, (x, y)) := y - \hat{y}, \tag{19}$$

where $\hat{y}$ is the prediction computed using the underlying algorithm U 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.) Similarly to the CPS based on Least Squares [21] and Kernel Ridge Regression [17] (as discussed above), this procedure is far from universal and can be expected to be efficient only for data that is not too far from being homoscedastic; this will be further discussed in Section 7 (see, in particular, Proposition 4).

Notice that, when implemented as in Algorithm 3, the SCPS is no longer provably calibrated (because parameter tuning in lines 16–17 depends on the full training sequence), and this is why we also check its validity in our experiments. To check the validity of both SCPS and CCPS, we run Algorithm 3 replacing $\text{CRPS}(F_i, y_i)$ with $F_i(y_i)$ and replacing boxplots with plots, such as those in Figure 7 (described in detail at the end of this section).

The `Boston Housing` dataset consists of 506 observations each with 14 attributes (describing an area of Boston) and a real-valued label (median house price in that area). Figure 2 shows the performance of the SCPS and CCPS.

The horizontal axis in the left panel is labelled by $\alpha \approx m/n$; the values of α used in our experiments are between 0.1 and 0.9, plus a few more extreme values. For a given value of α we set $m := \lfloor \alpha n \rfloor$. The CRPS loss is computed for the (crisp) SCPS based on (19) and the three underlying algorithms on each observation in the test sequence; as described above, we then represent the resulting 1000 CRPS losses as a boxplot. We can see a characteristic U-shape (especially pronounced on the left); small m/n lead to a significant increase in the CRPS loss, and large m/n lead to a slight increase in the CRPS loss but a significant increase in its variability (the rightmost box and its whiskers tend to be longer).

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.

The `Diabetes` dataset consists of 10 physiological measures on 442 patients, and the label indicates disease progression after one year. Figure 3 is the

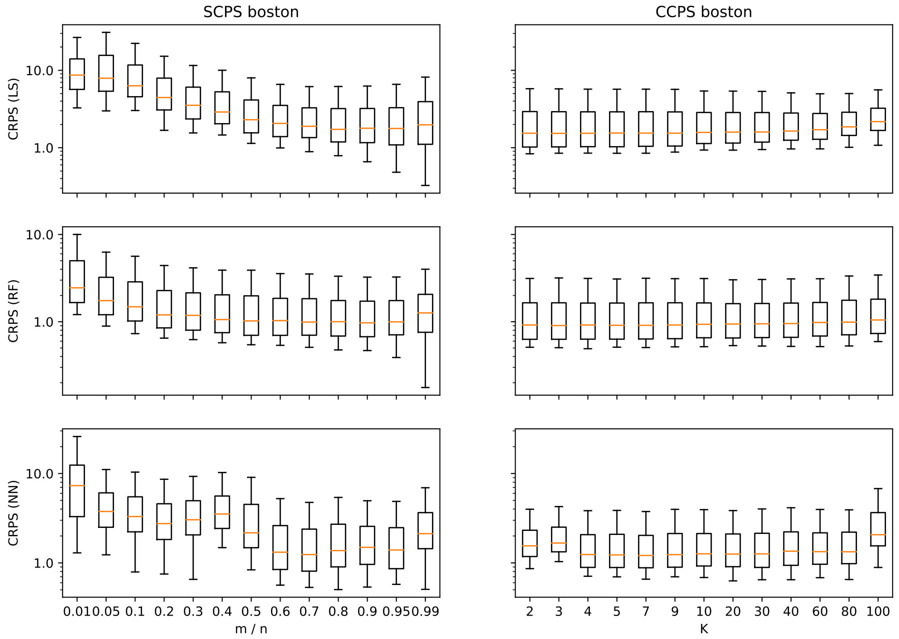

Figure 2: The performance of the SCPS (left panel) and CCPS (right panel) on the `Boston Housing` dataset using Least Squares (LS), Random Forest (RF), and Neural Networks (NN) as the underlying algorithms, as indicated on the left. The vertical axis uses the log scale and gives the CRPS. Left panel: the numbers on the horizontal axis are the fractions m/n of the training sequence used as the training sequence proper. Right panel: the numbers on the horizontal axis are the numbers K of folds.

analogue of Figure 2 for this dataset. We can see the same tendencies, with $K \in \{5, 10\}$ still being reasonable numbers of folds for CCPS.

The `Yacht Hydrodynamics` is the smallest of our datasets. It consists of 7 attributes including the basic hull dimensions and the boat velocity for 308 experiments, and the task is to predict the residuary resistance of sailing yachts. Figure 4 suggests that the behavior shown in Figures 2 and 3 is in fact typical of small datasets.

The `Wine Quality` dataset has information about 1599 red wines and 4898 white wines. We merge these two groups creating another attribute taking two values, 0 for white and 1 for red. The label is the quality of wine expressed as a score between 0 and 10. (The most common labels are 5 and 6, labels 3 and 9 are very uncommon, and labels 0 and 1 are absent.)

Figure 5 is qualitatively similar to Figures 2 and 3. The shape of the plots for SCPS suggests that we need a reasonable length $n - m$ of the calibration sequence, such as 100 or 200, since it determines the granularity of the pre-

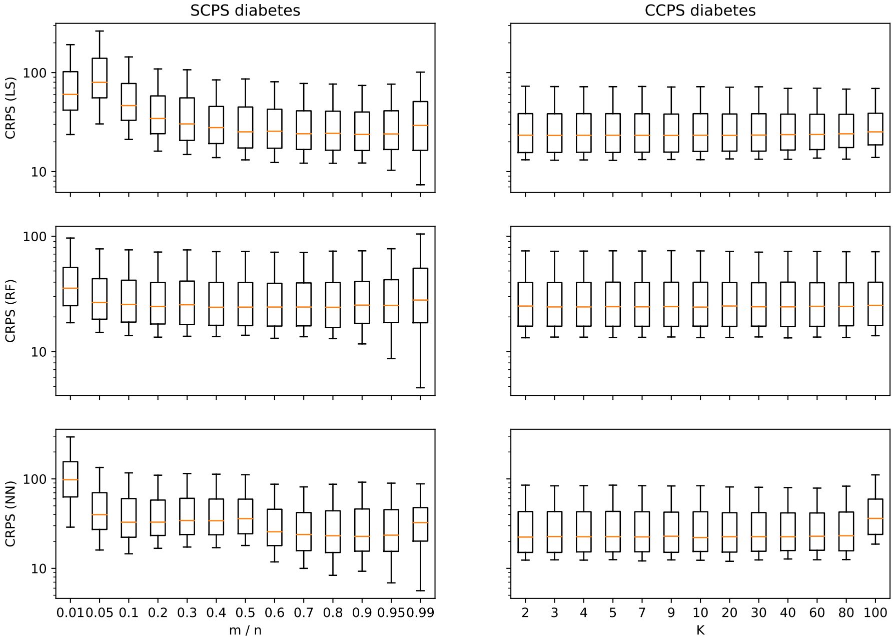

Figure 3: The analogue of Figure 2 for the `Diabetes` dataset.

dictive distributions: as we have already mentioned in connection with (9), the thickness of the predictive distribution is $\frac{1}{n-m+1}$. Increasing the length of the calibration sequence further does not improve the predictive performance significantly, and starts hurting it when the training sequence proper becomes too short.

Figure 6 reports the results for the largest dataset that we use, `Naval Propulsion`. It contains information about 11,934 simulated experiments, each described by 16 attributes, and the task is to predict the Gas Turbine Compressor decay state coefficient for a propulsion plant. Here we observe the same general behavior.

The best results presented in Figures 2–6 are summarized in Table 1. Namely, the table reports the median CRPS losses shown in Figures 2–5 obtained by optimizing the parameters m/n in the case of SCPS and K in the case of CCPS. In the majority of cases CCPS perform better than SCPS. But what is even more important, CCPS are much less sensitive to choosing their parameter K, and so the best results given in Table 1 are in fact typical for them. In all our experiments, it is safe to choose any of the standard values for the number K of folds in the range from 5 to 10.

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 [13,

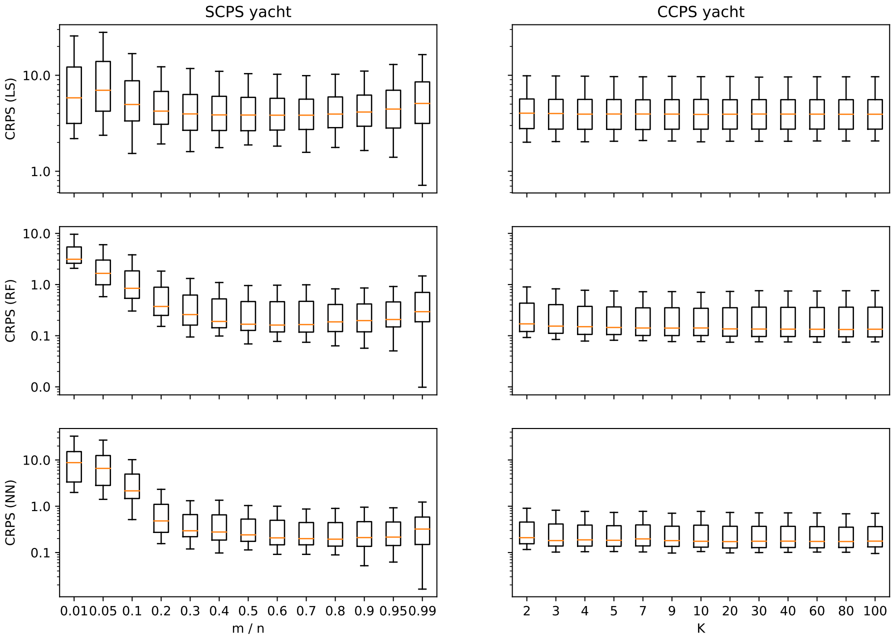

Figure 4: The analogue of Figure 2 for the `Yacht Hydrodynamics` dataset.

Appendix A] and experimentally [6] that a loss of validity is possible. Figure 7 (right panel) shows the distribution of the values (18) for `Boston Housing` and $K = 5$, where $z_1, \dots, z_n$ is the training sequence, and (x, y) range over the elements of the test sequence. 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 $Q(z_1, \dots, z_n, (x, y))$ for (x, y) in the test sequence that do not exceed α. The right panels of Figures 8, 9, 10, and 11 are the analogues for the `Diabetes`, `Yacht Hydrodynamics`, `Wine Quality`, and `Naval Propulsion` datasets, respectively. Under perfect validity (4) and an infinitely long test sequence, the calibration curves should be the diagonals shown as dashed lines on both panels of Figures 7–11; the actual calibration curves are fairly close. The calibration curves for other K are roughly similar. As mentioned earlier, we also give calibration results for SCPS (in the left panels and with $m/n \approx 0.5$).

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 (17) (in the case of SCPS) or (18) (in the case of CCPS) instead of (9) or (13), respectively. The CCPS is still empirically valid in our experiments, even in the extreme case of $K = 100$. On the other hand, when using (17) in place of (9), the SCPS lose not only theoretical but

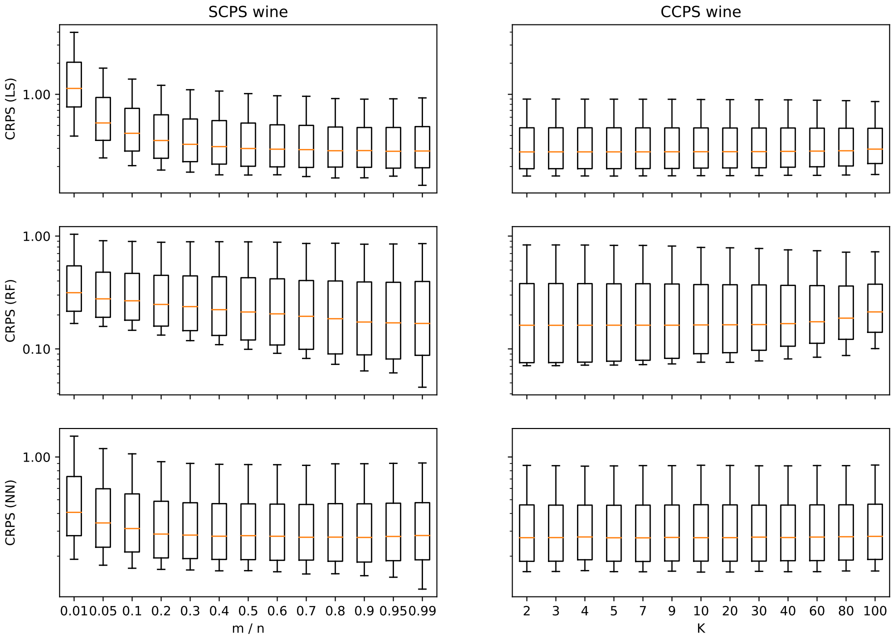

Figure 5: The analogue of Figure 2 for the `Wine Quality` dataset.

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 length of the calibration sequence is 4, and so the empirical predictive distribution (17) 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 Universal consistency of predictive systems

The conference version [16] of this paper was published in the proceedings of COPA 2018, which also contained a paper [8] that adapted Venn prediction for producing predictive distributions. In this and next sections we will analyze the asymptotic performance of the two approaches to predictive distributions, using conformal prediction [16] and using Venn prediction [8]. Our conclusion is that, as implemented in those papers, both approaches are very restrictive. But whereas the approach based on conformal prediction can be easily rescued, fixing the approach based on Venn prediction might require sacrificing computational efficiency. In this section we discuss the former approach.

Informally, an RPS is universally consistent if it gives the true predictive distribution in the limit, and a class of RPS is universal if it contains such an RPS. The following formalization is given in [12] and its idea goes back to

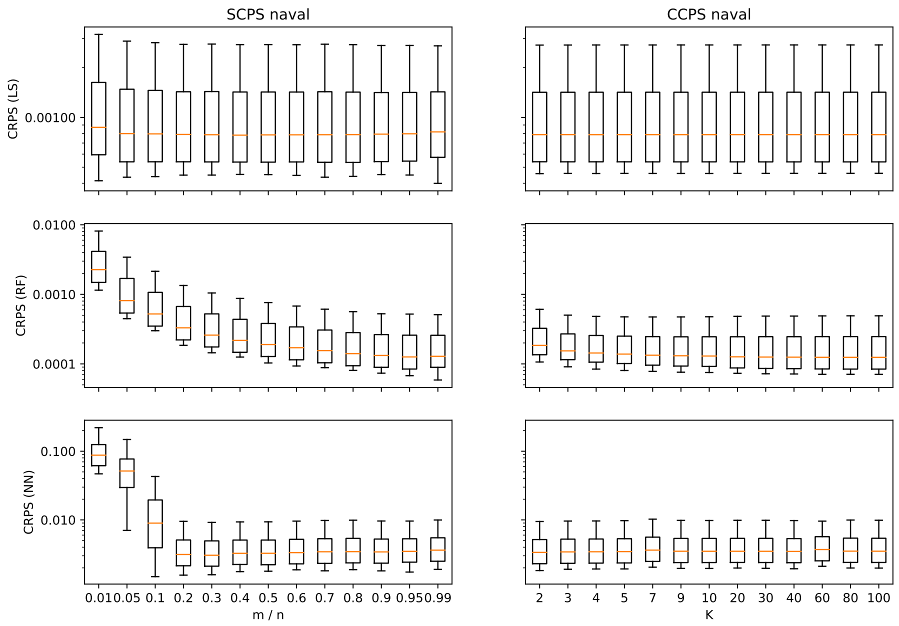

Figure 6: The analogue of Figure 2 for the `Naval Propulsion` dataset.

Belyaev's work (see, e.g., [2]).

Definition 2. An RPS Q is *consistent* for a probability measure P on $\mathbf{Z}$ if, for any bounded continuous function $f : \mathbb{R} \to \mathbb{R}$,

$$\int f \, \mathrm{d}Q_n - \mathbb{E}(f \mid x_{n+1}) \to 0 \qquad (n \to \infty) \tag{20}$$

in probability, where:

- Q_n is the predictive distribution function (5) for the label of x_{n+1} based on the training sequence $z_1, \dots, z_n$; the integral $\int f \, \mathrm{d}Q_n$ is not quite standard since we did not require Q_n to be exactly a distribution function, and we understand it as $\int f \, \mathrm{d}\bar{Q}_n$ with the measure $\bar{Q}_n$ on $\mathbb{R}$ defined by $\bar{Q}_n((u,v]) := Q_n(v+) - Q_n(u+)$ for any interval $(u,v]$ of this form in $\mathbb{R}$;

- $\mathbb{E}(f \mid x_{n+1})$ is the conditional expectation for $f(y)$ given $x = x_{n+1}$ assuming $(x,y) \sim P$ (we fix a version of the conditional expectation);

- the data-generating and coin-tossing mechanisms are $z_i = (x_i, y_i) \sim P$, $i = 1, \dots, n+1$, and $\tau \sim U$, assumed all independent.

We say that Q is *universally consistent* if it is consistent for any probability measure P on $\mathbf{Z}$. A class of RPS is *universal* if it contains a universally consistent RPS.

Table 1: Best results for the median CRPS loss for SCPS and CCPS for the five datasets and three underlying algorithms.

Dataset	underlying algorithm	SCPS	CCPS
Boston Housing	Least Squares	1.726	1.533
Boston Housing	Random Forest	0.972	0.906
Boston Housing	Neural Network	1.240	1.211
Diabetes	Least Squares	23.74	23.18
Diabetes	Random Forest	24.23	24.33
Diabetes	Neural Network	22.76	22.10
Yacht Hydrodynamics	Least Squares	3.840	3.910
Yacht Hydrodynamics	Random Forest	0.1615	0.1322
Yacht Hydrodynamics	Neural Network	0.1944	0.1725
Wine Quality	Least Squares	0.2810	0.2771
Wine Quality	Random Forest	0.1681	0.1618
Wine Quality	Neural Network	0.2711	0.2693
Naval Propulsion	Least Squares	0.0007812	0.0007866
Naval Propulsion	Random Forest	0.0001259	0.0001242
Naval Propulsion	Neural Network	0.003051	0.003360

The requirement of a class of RPS being universal means that it does not impose insurmountable limits to getting the data-generating distribution right.

We will also apply Definition 2 to predictive systems that are not required to satisfy the validity condition R2 in Definition 1 (such as CCPS and predictive systems based on Venn prediction). The following theorem assumes that the notions of SCPS and CCPS have been slightly extended by allowing randomized conformity measures (see, e.g., [12, Section 9] for details).

Theorem 3. *The class of SCPS is universal. The class of CCPS is also universal.*

Proof. The class of SCPS being universal is a simplified version of [12, Theorem 31]. For a fixed K, a K-fold CCPS outputs predictive distribution functions within $O(1/n)$ of the average of the predictive distribution functions output by the component SCPS (see (15)), which immediately implies that the class of CCPS is also universal. □

Theorem 3 says that, in principle, conformal predictive systems can adapt to any data-generating distribution. However, specific conformal predictive systems considered in literature are often not universally consistent. This is particularly true for predictive systems based on full (rather than split or cross-) conformal prediction, where computational efficiency imposes severe restrictions on the underlying algorithm.

As discussed earlier, an example of a non-universal class of RPS is provided by [21]: they are based on the method of Least Squares and therefore far from

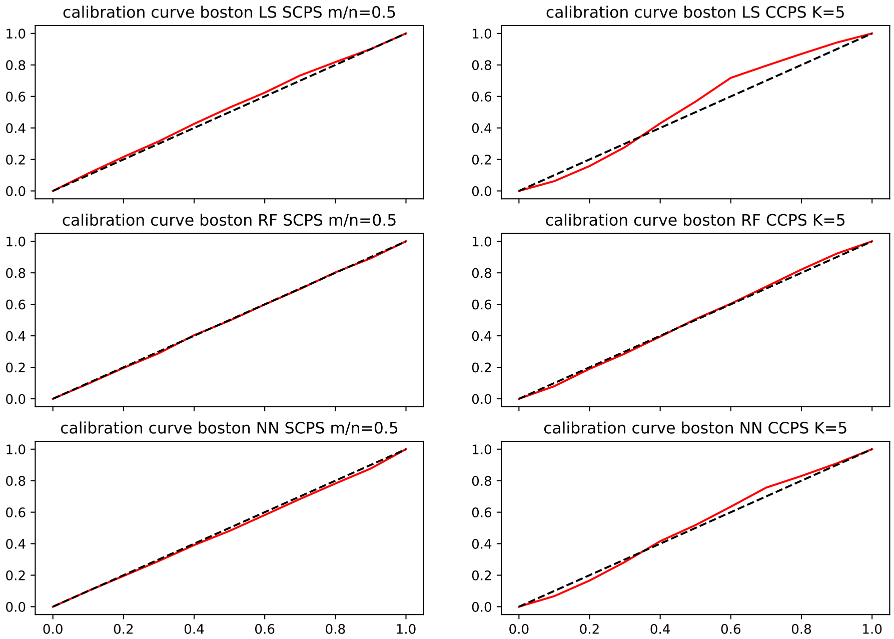

Figure 7: The calibration curves (i.e., the distributions of $Q(z_1, \ldots, z_n, (x, y))$ over the test sequence) for the SCPS and CCPS on the `Boston Housing` dataset.

being universal. The extension of Least Squares to Kernel Ridge Regression given in [17] still does not produce universality, even for universal kernels: the Kernel Ridge Regression Prediction Machine introduced in [17] is not universal since the shape of its predictive distributions is not tailored to a specific test observation [17, Section 7].

Conformal predictive systems based on (10) are also not universal: they allow any shape of the asymptotic predictive distribution function, but this shape is adapted to the test object at hand only by shifting and scaling it (by *shifting* a distribution function F we mean replacing it by the distribution function $y \mapsto F(y - c)$ for some $c \in \mathbb{R}$, and by *scaling* we mean replacing it by $y \mapsto F(y/\sigma)$ for some $\sigma > 0$). The class (19) of split and cross-conformity measures considered in the experimental section is even more restrictive: the asymptotic shape of the predictive distribution function is adapted to the test object at hand only by a shift. We will state in detail only the claim about the class (19), since it was the main class used in this paper, and it is also the class used in [21] and [17].

Let us call, for want of a better name, split conformity measures of the form (19) *simple*. Suppose the probability measure P generating the observations (x, y) satisfies $\mathbb{E}\,|y| < \infty$ for $(x, y) \sim P$. Set $A(x, y) := y - \hat{y}$, where $\hat{y} := \mathbb{E}(y \mid x)$ is a fixed version of the conditional expectation of y given x for $(x, y) \sim P$. The

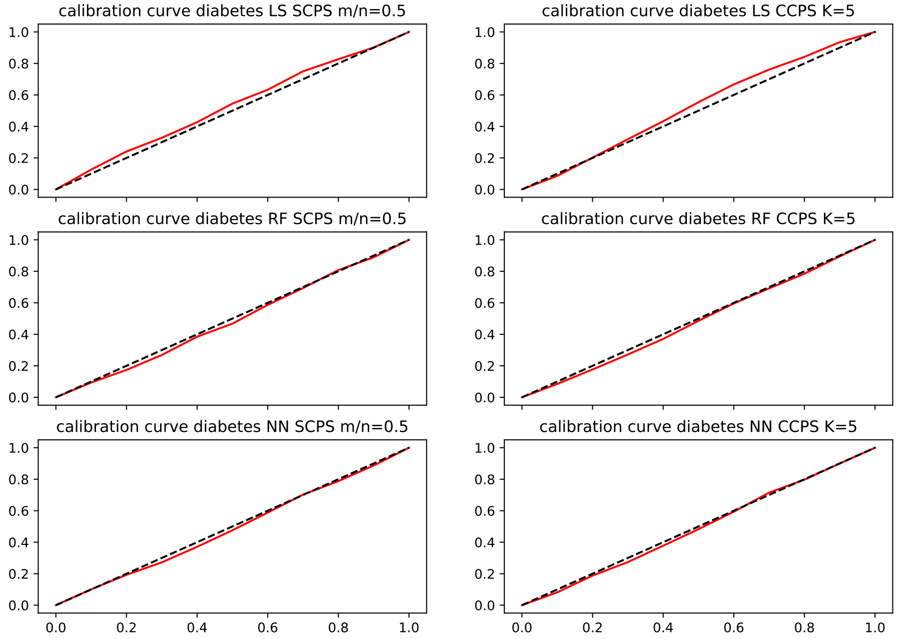

Figure 8: The analogue of Figure 7 for the `Diabetes` dataset.

ideal simple conformal predictive system (ISCPS) for P is defined as

$$Q(z_1,\dots,z_n,(x,y),\tau) := \frac{1}{n+1}\left|\{i=1,\dots,n \mid A(x_i,y_i) < A(x,y)\}\right| \\ + \frac{\tau}{n+1}\left|\{i=1,\dots,n \mid A(x_i,y_i) = A(x,y)\}\right| + \frac{\tau}{n+1}, \quad (21)$$

where x is the test object. The intuition behind this definition is that we are given P in advance and, therefore, can use the whole training sequence as the calibration sequence; a training sequence proper is not needed as A is already the ideal simple conformity measure. An ISCPS is an idealization of SCPS corresponding to an infinitely long training sequence proper (allowing a perfect estimate of $\mathbb{E}(y \mid x)$). Since CCPS are essentially combinations of SCPS, our conclusions will also be applicable to CCPS.

Remember that the Kolmogorov distance between distribution functions F and G is

$$K(F,G) := \sup_{u\in\mathbb{R}} \left|F(u) - G(u)\right|.$$

Modify it by setting

$$K'(F,G) := \inf_{c\in\mathbb{R}} \sup_{u\in\mathbb{R}} \left|F(u-c) - G(u)\right|.$$

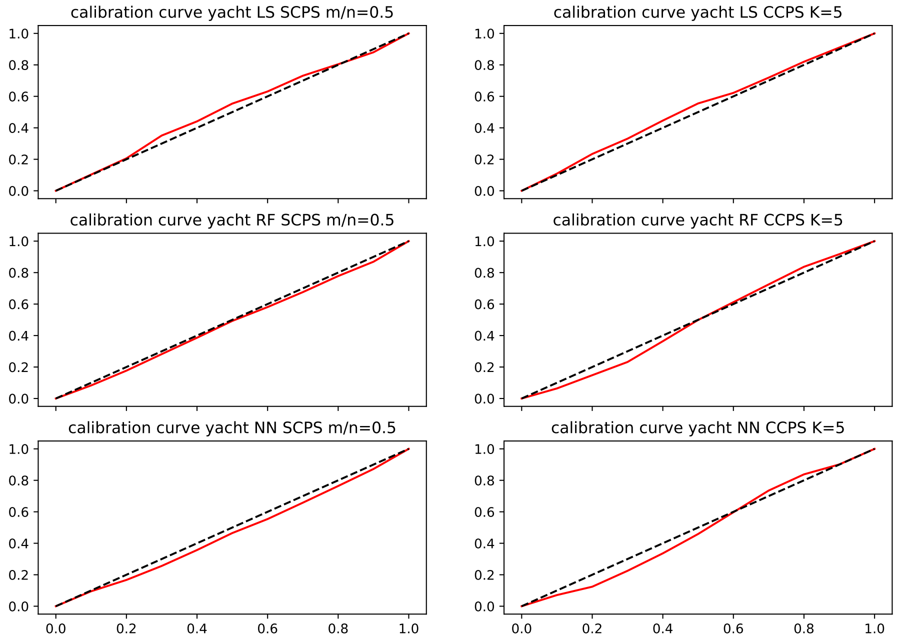

Figure 9: The analogue of Figure 7 for the `Yacht Hydrodynamics` dataset.

This is not a metric anymore: $K'(F,G) = 0$ only means that F and G coincide to within a shift left or right. The following proposition spells out the observation above that, in the case of a simple conformity measure, the asymptotic shape of the predictive distribution function is adapted to the test object at hand only by a shift.

Proposition 4. *Let Q be an ISCPS. For all n, all $z_1, \dots, z_n \in \mathbf{Z}$, all $x, x' \in \mathbf{X}$, and all $\tau \in [0,1]$,*

$$K'\left(Q(z_1,\dots,z_n,(x,\cdot),\tau), Q(z_1,\dots,z_n,(x',\cdot),\tau)\right) = 0.$$

Proof. Since $A(x,y) = y - \hat{y}$, $Q(z_1,\dots,z_n,(x,\cdot),\tau)$ is of the form $F(\cdot - \hat{y})$ and $Q(z_1,\dots,z_n,(x',\cdot),\tau)$ is of the form $F(\cdot - \hat{y}')$ for some numbers $\hat{y}$ and $\hat{y}'$ and some function F (see (21)). Therefore, they are shifts of each other. □

Proposition 4 will remain true if $\hat{y} := \mathbb{E}(y \mid x)$ in the definition of a simple conformity measure is replaced by $\hat{y} := f(x)$ for any function $f : \mathbf{X} \to \mathbb{R}$.

The idealized version of the split conformity measure (10) is

$$A(x,y) := \frac{y - f(x)}{\sigma(x)} \tag{22}$$

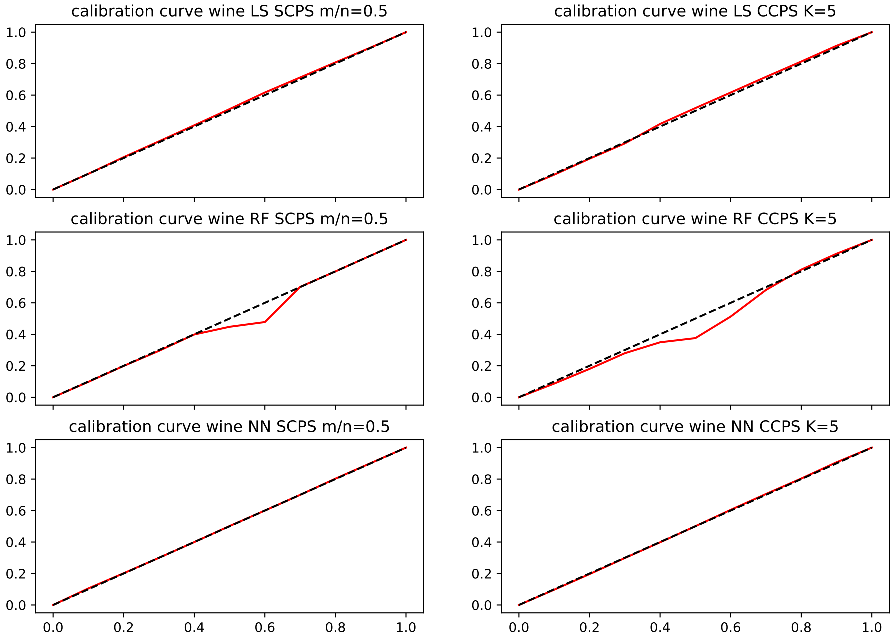

Figure 10: The analogue of Figure 7 for the `Wine Quality` dataset.

for some positive function $\sigma : \mathbf{X} \to (0, \infty)$. Let us modify further the Kolmogorov distance by setting

$$K''(F, G) := \inf_{c \in \mathbb{R}, \sigma > 0} \sup_{u \in \mathbb{R}} \left| F\left(\frac{u - c}{\sigma}\right) - G(u) \right|.$$

Proposition 4 will continue to hold if we allow ISCPS to use idealized split conformity measures (22) and replace K' by K''.

Using split and cross-conformal predictive systems rather than full conformal predictive systems makes it much easier to design adaptive conformity measures. One possibility is to use the Nadaraya–Watson estimate (introduced by [7] and [23] in the case of regression and [9] in the case of density estimation)

$$F(y \mid x) = \frac{\sum_{i=1}^{m} \Sigma\left(\frac{y - y_i}{h_y}\right) K\left(\frac{x - x_i}{h_x}\right)}{\sum_{i=1}^{m} K\left(\frac{x - x_i}{h_x}\right)} \tag{23}$$

of the conditional distribution function for computing the conformity score of (x, y) given $(x_1, y_1), \dots, (x_m, y_m)$). The parameters of the estimator (23) are a distribution function Σ (e.g., the Heaviside step function or a smooth one, such as the sigmoid $\Sigma(u) := 1/(1 + e^{-u})$, in which case there is a unique solution to the equations in Algorithms 1 and 2), a kernel K (such as the Gaussian

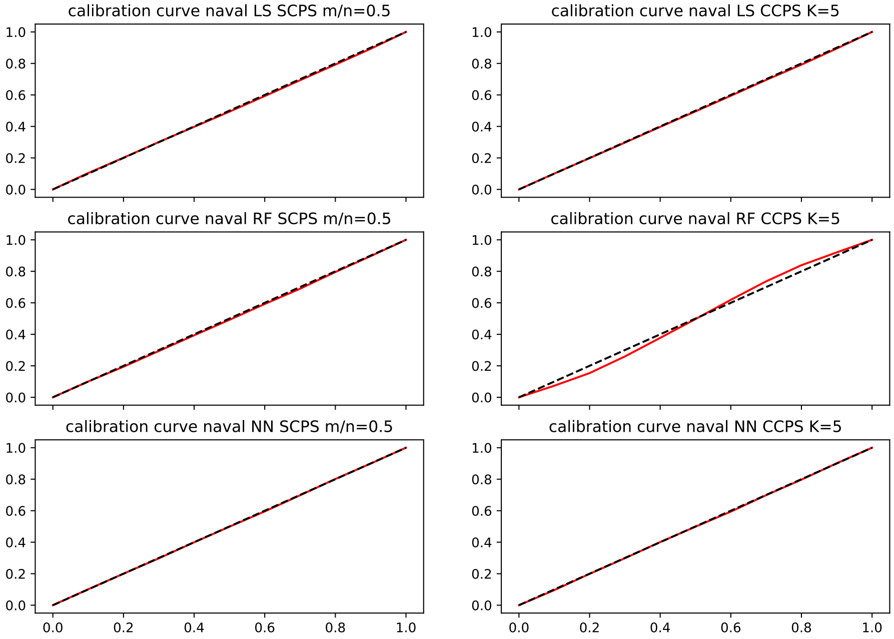

Figure 11: The analogue of Figure 7 for the `Naval Propulsion` dataset.

$K(u) := \exp(-u^2/2)$), and bandwidths $h_x > 0$ and $h_y > 0$. This is the topic of [20].

8 Split Venn–Abers predictive systems

In this section we discuss an alternative to RPS introduced in [8] and based on Venn prediction. We will obtain a modification of RPS defined as follows (cf. Definition 1).

Definition 3. A function $Q : \mathbf{Z}^{n+1} \times \{0,1\} \to [0,1]$ is called an *imprecise predictive system* (IPS) if it satisfies the following two requirements:

i For each training sequence $(z_1,\ldots,z_n) \in \mathbf{Z}^n$ and each test object $x \in \mathbf{X}$, the function $Q(z_1,\ldots,z_n,(x,y),\tau)$ is monotonically increasing both in y and in τ. In other words, for either $\tau \in \{0,1\}$, the function (1) is monotonically increasing, and for each $y \in \mathbb{R}$,

$$Q(z_1,\ldots,z_n,(x,y),0) \le Q(z_1,\ldots,z_n,(x,y),1).$$

ii For each training sequence $(z_1,\ldots,z_n) \in \mathbf{Z}^n$ and each test object $x \in \mathbf{X}$, we have (2) and (3).

As compared with Definition 1, we drop the validity condition R2 (which may be stated separately in some form when needed).

We start the definition of a Venn-type IPS from an analogue of a conformity measure.

Definition 4. A *regressor* is a family of measurable functions $A_m : \mathbf{Z}^m \times \mathbf{X} \to \mathbb{R}$, $m = 1, 2, \ldots$.

The intention is that $A_m(z_1, \ldots, z_m, x)$ is the prediction for the label of x computed from $z_1, \ldots, z_m$ as training sequence. As before, we drop the lower index m in A_m. Now we can define a new kind of predictive systems.

Definition 5. Split the training sequence $z_1, \ldots, z_n$ into two parts: the *training sequence proper* $z_1, \ldots, z_m$ and the *calibration sequence* $z_{m+1}, \ldots, z_n$. Suppose we are given a test object x and a possible label $y \in \mathbb{R}$ for it. The output $Q(z_1, \ldots, z_n, (x, y), \tau)$, $\tau \in \{0, 1\}$, of the *split Venn–Abers predictive system of type* T determined by the regressor A, where $T \in \{1, 2, 3\}$, is defined as follows:

- set $s_i := A(z_1, \ldots, z_m, x_i)$ for $i = 1, \ldots, n$ and set $s := A(z_1, \ldots, z_m, x)$;
- fit an isotonic regression $g : \mathbb{R} \to \mathbb{R}$ to the training sequence (s_i, y_i^*) extended by adding (s, τ), where

  $$y_i^* := \begin{cases} 0 & \text{if } y_i \le y \\ 1 & \text{otherwise} \end{cases} \tag{24}$$

  and the range of i is

  $$i = \begin{cases} m+1, \ldots, n & \text{if } T = 1 \\ 1, \ldots, m & \text{if } T = 2 \\ 1, \ldots, n & \text{if } T = 3; \end{cases}$$

  the corresponding optimization problem is

  $$(\tau - g(s))^2 + \sum_i (y_i^* - g(s_i))^2 \to \min \tag{25}$$

  under the restriction that g is monotonically increasing;
- set

  $$Q(z_1, \ldots, z_n, (x, y), \tau) := 1 - \frac{|\{i = m+1, \ldots, n \mid g(s_i) = g(s), y_i^* = 1\}| + \tau}{|\{i = m+1, \ldots, n \mid g(s_i) = g(s)\}| + 1}. \tag{26}$$

An IPS is a *split Venn–Abers predictive system* (SVAPS) of type T if it is the split Venn–Abers predictive system determined by some regressor.

Intuitively, a SVAPS tries to answer the question whether the label of the test object x exceeds y based on the answers (24) for training objects; the answer is given by the fraction in (26). Notice that:

- SVAPS satisfy the definition of an IPS. Indeed, suppose, e.g., that y is sufficiently large (the case $y \to \infty$). Then we will have $y_i^* = 0$ for all i. For $\tau = 0$ we will have $g = 0$ and the fraction in (26) will be 0. The argument for the case $y \to -\infty$ is analogous.
- The integral $\int f \, \mathrm{d}Q_n$ in (20) may depend on τ, and in the definitions of consistency and universality for SVAPS we require that (20) hold for either value of τ.

SVAPS of type 1 were introduced in [8, Section 3.1.2] as the most direct application of the Venn–Abers methodology [18, 19] to the problem of probabilistic regression. Since the arguments s_i and s of the function g in (26) are also used in the condition (25), the expressions $g(s_i)$ and $g(s)$ in (26) are determined uniquely; therefore, the definition (26) of type 1 SVAPS is unambiguous.

SVAPS of type 2 were introduced in [8, Section 3.1.3] as a computational simplification of SVAPS of type 1; additionally, [8, Section 3.1.3] removes the first addend in (25) (which is the key step in achieving computational efficiency). For this version of SVAPS, the arguments s_i and s of the function g in (26) are not necessarily used in the condition (25), and so the expressions $g(s_i)$ and $g(s)$ in (26) may not be determined uniquely. Somewhat arbitrarily, we may define $g(t)$, for any $t \in \mathbb{R}$, as $g(s_j)$ where s_j is the nearest neighbour to t among s_i, $i \in \{1, \dots, m\}$; in the case of ties, we choose j as small as possible. This makes the definition (26) of type 2 SVAPS also unambiguous.

SVAPS of type 3 are a natural combination of SVAPS of types 1 and 2; they are similar to SVAPS of type 1 in that the definition (26) is for them unambiguous.

The validity guarantees for SVAPS are very different from those that we have for CPS and SCPS; see, e.g., [12, Appendix B].

The following simple example illustrates severe restrictions of SVAPS (of any type), even when the training sequence is very long, stemming from the score $A(z_1, \dots, z_m, x)$ being just one number.

Example 5. The true distribution generating the observations (x, y) produces $x = 0$ and $x = 1$ with equal probabilities. Given $x = 0$, we have $y = 0$ with probability 1. Given $x = 1$, we have $y = -1$ or $y = 1$ with equal probabilities. Let us check that SVAPS are not consistent, even for the ideal regressor $A(z_1, \dots, z_m, x) := 0$. In the notation of Definition 5, we have $s = 0$ and $s_i = 0$ for $i = 1, \dots, n$. The asymptotic predictive distribution is shown in Figure 12, concentrated on $\{-1, 0, 1\}$, and assigns probabilities 1/4, 1/2, and 1/4 to -1, 0, and 1, respectively. It is very poor; the expected CRPS (as defined in Section 5) for it is 3/8 instead of the ideal 1/4.

Example 5 makes it plausible that the SVAPS are not universal; now we state this formally.

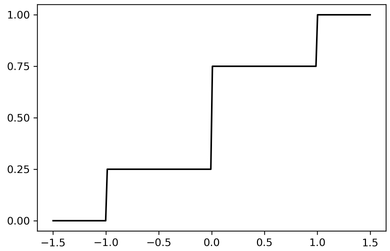

Figure 12: The asymptotic predictive distribution produced by SVAPS for any x in Example 5.

Proposition 6. *The SVAPS are not universal.*

Proof. Let the true probability measure be the one described in Example 5. Let $n \to \infty$ and set $m := \lfloor n/2 \rfloor$. Set

$$\begin{aligned} A_0 &:= A(z_1, \dots, z_m, 0) \\ A_1 &:= A(z_1, \dots, z_m, 1) \end{aligned}$$

(in Example 5 we only considered the case $A_1 = A_0 = 0$). If $A_1 = A_0$, we are in the situation of Example 5; see Figure 12. For a continuous $f : \mathbb{R} \to [0, 1]$ satisfying

$$f(u) = \begin{cases} 1 & \text{if } y \geq 0.6 \\ 0 & \text{if } y \leq 0.4 \end{cases} \tag{27}$$

we will have

$$\lim_{\substack{x_{n+1}=0 \\ n\to\infty}} \int f \, \mathrm{d}Q_n = \frac{1}{4} \neq 0 = \lim_{\substack{x_{n+1}=0 \\ n\to\infty}} \mathbb{E}(f \mid x_{n+1}) \qquad \text{a.s.} \tag{28}$$

and

$$\lim_{\substack{x_{n+1}=1 \\ n\to\infty}} \int f \, \mathrm{d}Q_n = \frac{1}{4} \neq \frac{1}{2} = \lim_{\substack{x_{n+1}=1 \\ n\to\infty}} \mathbb{E}(f \mid x_{n+1}) \qquad \text{a.s.} \tag{29}$$

If $A_1 < A_0$, the predictive distributions are as shown in Figure 13 (the weights for -1, 0, and 1 are 0, 3/4, and 1/4, respectively, when $x = 0$, and 1/2, 1/4, and 1/4, respectively, when $x = 1$). Taking the same function f, satisfying (27), we will still have (28) and (29).

If $A_0 < A_1$, the predictive distributions are as shown in Figure 14 (the weights for -1, 0, and 1 are 1/4, 3/4, and 0, respectively, when $x = 0$, and 1/4, 1/4, and 1/2, respectively, when $x = 1$). For a continuous $f : \mathbb{R} \to [0, 1]$ satisfying

$$f(u) = \begin{cases} 1 & \text{if } y \leq -0.6 \\ 0 & \text{if } y \geq -0.4 \end{cases}$$

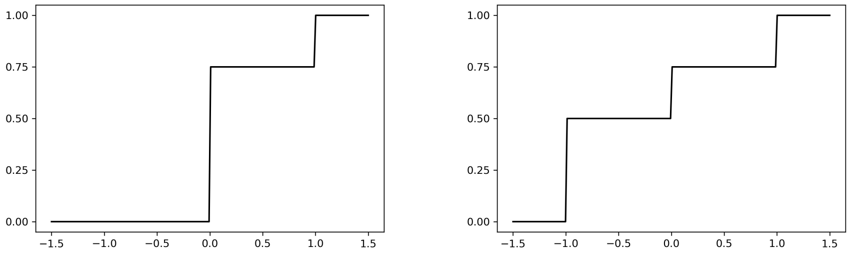

Figure 13: The asymptotic predictive distributions produced by SVAPS when $A_1 < A_0$ for $x = 0$ (left panel) and $x = 1$ (right panel).

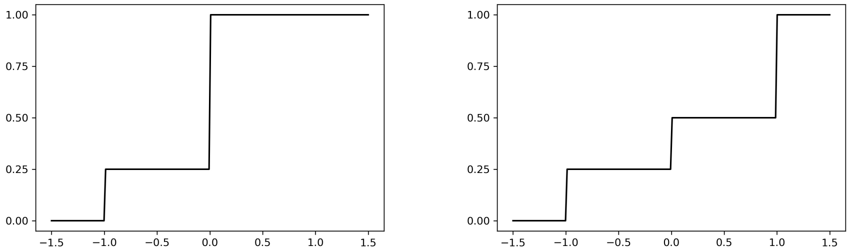

Figure 14: The asymptotic predictive distributions produced by SVAPS when $A_0 < A_1$ for $x = 0$ (left panel) and $x = 1$ (right panel).

we will still have (28) and (29). □

Remark 1. In the proof of Proposition 6 we checked that the SVAPS are not universal directly, using the definition (20). This shows in their inferior CRPS, as in Example 5. If $A_1 = A_0$, we are in the situation of Example 5. If $A_1 < A_0$, the expected CRPS is 5/16, which exceeds the ideal value 1/4. And if $A_0 < A_1$, the expected CRPS is also 5/16.

The asymptotic problem of non-universality for SVAPS can be avoided by modifying, for each $y \in \mathbb{R}$, the data sequence (including the training sequence proper) as follows: replace each y_i by $\mathbf{1}_{\{y_i \le y\}}$. There are, however, two problems with this procedure:

- Loss of computational efficiency; now processing even moderately large datasets becomes infeasible.
- Loss of predictive efficiency for small samples as now the labels become less informative (only taking values 0 or 1).

In this section we have only discussed SVAPS in detail, but the same argument shows that cross-Venn–Abers predictive systems (defined in a natural way) inherit the lack of universality.

9 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 five benchmark datasets and three underlying algorithms. 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 datasets and underlying algorithms.

The specific split and cross-conformity measures used in this paper, all of which have the form (19), are not fully adaptive, as discussed in Section 7, whereas in general SCPS and CCPS are universal (unlike SVAPS, as pointed out in Section 8). Replacing (19) by the more general (10) somewhat improves the attainable flexibility, but designing fully flexible cross-conformal predictive systems based on efficient non-parametric predictive systems, such as the Nadaraya–Watson system (23), appears to us a particularly interesting direction of further research.

Acknowledgments

Thanks to the AstraZeneca team (first of all Claus Bendtsen and Ola Engkvist) for useful discussions and to Wang Di of Tianjin University for correcting an error. We are very grateful to the three anonymous referees of the conference version of this paper for their thoughtful comments; because of time limitations we could not fully take some of them into account in the conference version but have done it in this journal version. Further comments by the three anonymous referees of the journal version are also gratefully appreciated; they have led to important improvements (such as performing more extensive computational experiments).

Funding: This work was supported by the EU Horizon 2020 Research and Innovation programme [grant number 671555]; AstraZeneca [grant number R10911, “Machine Learning for Chemical Synthesis”]; Leverhulme Magna Carta Doctoral Centre at Royal Holloway, University of London; and NHS England supported by Innovate UK [Technology Integrated Health Management project awarded to the School of Mathematics and Information Security at Royal Holloway, University of London].

References

[1] Vineeth N. Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk, editors. *Conformal Prediction for Reliable Machine Learning: Theory, Adap-*

tations, and Applications. Elsevier, Amsterdam, 2014.

[2] Yuri Belyaev and Sara Sjöstedt–de Luna. Weakly approaching sequences of random distributions. *Journal of Applied Probability*, 37:807–822, 2000.

[3] 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.

[4] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

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

[6] 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.

[7] Elizbar A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[8] Ilia Nouretdinov, Denis Volkhonskiy, Pitt Lim, Paolo Toccaceli, and Alexander Gammerman. Inductive Venn–Abers predictive distribution. *Proceedings of Machine Learning Research*, 91:15–36, 2018. COPA 2018.

[9] Murray Rosenblatt. Conditional probability density and regression estimators. In Paruchuri R. Krishnaiah, editor, *Multivariate Analysis II*, pages 25–31. Academic Press, New York, 1969.

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

[11] 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.

[12] Vladimir Vovk. Universally consistent predictive distributions, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 18, August 2019 (first posted April 2017).

[13] Vladimir Vovk. Cross-conformal predictors, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 6, January 2013 (first posted August 2012).

[14] Vladimir Vovk and Claus Bendtsen. Conformal predictive decision making, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 19, December 2018 (first posted July 2017).

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

[16] Vladimir Vovk, Ilia Nouretdinov, Valery Manokhin, and Alex Gammerman. Cross-conformal predictive distributions. *Proceedings of Machine Learning Research*, 91:37–51, 2018. COPA 2018.

[17] Vladimir Vovk, Ilia Nouretdinov, Valery Manokhin, and Alex Gammerman. Conformal predictive distributions with kernels, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 20, January 2019 (first posted October 2017).

[18] Vladimir Vovk and Ivan Petej. Venn–Abers predictors, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 7, June 2014 (first posted October 2012).

[19] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 13, November 2019 (first posted November 2015).

[20] Vladimir Vovk, Ivan Petej, Paolo Toccaceli, and Alex Gammerman. Conformal calibrators, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 23, February 2019.

[21] Vladimir Vovk, Jieli Shen, Valery Manokhin, and Minge Xie. Nonparametric predictive distributions based on conformal prediction, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 17, March 2019 (first posted April 2017).

[22] Vladimir Vovk and Ruodu Wang. Combining p-values via averaging, On-line Compression Modelling project (New Series), `http://alrw.net`, Working Paper 21, November 2019 (first posted November 2017).

[23] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā A*, 26:359–372, 1964.

Algorithm 3 Experiments in Section 6

```
Require: A dataset of size $n + l$ consisting of observations $z = (x, y)$, $y \in \mathbb{R}$.
for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
    for $\alpha \in \{0.01, 0.05, \dots, 0.99\}$ do
        Create an empty multiset $B_{U,\alpha}$.
    end for
end for
for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
    for $K \in \{2, 3, \dots, 100\}$ do
        Create an empty multiset $B'_{U,K}$.
    end for
end for
for 10 times do
    Randomly permute the dataset obtaining a sequence $z_1, \dots, z_{n+l}$.
    Use $z_1, \dots, z_n$ as the training and $z_{n+1}, \dots, z_{n+l}$ as the test sequence.
    Apply feature scaling by fitting on the training sequence and transforming the training and test sequences.
    Tune the parameters using 3-fold cross-validation on the training sequence.
    for $\alpha \in \{0.01, 0.05, \dots, 0.99\}$ do
        $m := \lfloor \alpha n \rfloor$.
        for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
            Train an SCPS based on $U$ using $z_1, \dots, z_m$ as training sequence proper and $z_{m+1}, \dots, z_n$ as calibration sequence.
            Put all $\mathrm{CRPS}(F_i, y_i)$, $i \in \{n + 1, \dots, n + l\}$ in $B_{U,\alpha}$, where $F_i$ is the output of the SCPS for $x_i$.
        end for
    end for
    for $K \in \{2, 3, \dots, 100\}$ do
        Put all $z_i$, $i \in \{\lceil (k-1)n/K \rceil + 1, \lceil kn/K \rceil\}$, into fold $k$, $k \in \{1, \dots, K\}$.
        for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
            Train a CCPS based on $U$ using these folds.
            Put all $\mathrm{CRPS}(F_i, y_i)$, $i \in \{n + 1, \dots, n + l\}$ in $B'_{U,K}$, where $F_i$ is the output of the CCPS for $x_i$.
        end for
    end for
end for
for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
    for $\alpha \in \{0.01, 0.05, \dots, 0.99\}$ do
        Show the multiset $B_{U,\alpha}$ as boxplot.
    end for
end for
for $U \in \{\mathrm{LS}, \mathrm{RF}, \mathrm{NN}\}$ do
    for $K \in \{2, 3, \dots, 100\}$ do
        Show the multiset $B'_{U,K}$ as boxplot.
    end for
end for
```