Machine Learning (2019) 108:489–510
https://doi.org/10.1007/s10994-018-5754-9

Combination of inductive mondrian conformal predictors

Paolo Toccaceli[1] · **Alexander Gammerman**[1]

Received: 16 February 2018 / Accepted: 1 August 2018 / Published online: 22 August 2018

Abstract

It is well known that ensembling predictions from different Machine Learning (ML) algorithms can improve accuracy. This paper proposes a approach to combine Conformal Predictors (CPs) with different underlying ML algorithms in a way that preserves their key property, i.e. validity. Different combination methods are discussed and their performance is evaluated on a chemoinformatics problem. In order to deal with the size, high-dimensionality, and strong imbalance of the data set, the paper applies a special type of CP: an Inductive Mondrian Conformal Predictor. We propose and evaluate, alongside methods from Statistical Hypothesis Testing, a heuristically motivated method for learning to combine CPs to improve the quality of prediction. We also explore a general nonparametric method for recovering validity after combination using a calibration set. On a real-world data set, several of the combined predictors consistently outperform the base CPs.

1 Introduction

Since their introduction (Vovk et al. 2005), Conformal Predictors (CP) have been applied in several fields as a principled way to obtain confidence predictions (Ivina et al. 2012; Lambrou et al. 2009; Laxhammar and Falkman 2010; Ji et al. 2008; Shabbir et al. 2015; Balasubramanian et al. 2009). The main attraction of CP is the theoretical guarantee on the (long-term) error rate of the predictions. Specifically, chosen an error rate ϵ, CP can output a *region prediction* that is guaranteed to contain the correct label with a relative frequency of $1 - \epsilon$. In addition to that, CP can output a confidence measure, i.e. a p value $p_y(\mathbf{x})$ for the hypothesis that the label of a test object $\mathbf{x}$ is y. This p value can be used for the combination of the predictions from multiple heterogeneous classifiers. In this paper we apply CPs and several combination methods to the problem of predicting the biological activity of chemical compounds. The problem is encountered in pharmaceutical drug discovery, where one possible approach is to screen a large number of compounds (possibly from a library) for activity towards a target of interest (e.g. an enzyme that is part of signalling path or metabolic path involved in a disaease). The ability to predict with a set level of

Editor: Lars Carlsson.

✉ Paolo Toccaceli
Paolo.Toccaceli@rhul.ac.uk

[1] Royal Holloway, University of London, Egham, Surrey, UK

confidence the activity of compounds is of extreme interest to the pharmaceutical industry as it can reduce significantly the investment required to identify a suitable number of "hits" for the subsequent stages of drug development. In the following section, we recall briefly the relevant concepts and definitions pertaining to Conformal Prediction, referring the reader to the existing literature (Gammerman and Vovk 2007; Shafer and Vovk 2008; Toccaceli et al. 2016) for the full treatment.

2 Conformal predictors

Assuming that the training set is made up of ℓ independents identically distributed examples (iid)[1] $(x_i, y_i) \in \mathbf{Z} = \mathbf{X} \times \mathbf{Y}$, if $x_{\ell+1}$ is a test example taken from the same distribution as the training examples, a Conformal Predictor assigns a p value to a hypothetical assignment of a label $y_{\ell+1}$ to the object $x_{\ell+1}$. Within the context of Conformal Prediction, the definition of p value relies on the notion of Non-Conformity Measure (NCM). The NCM is a real-valued function $A(z; \wr z_1, \ldots, z_k \int)$, $A : \mathbf{Z} \times \mathbf{Z}^{(k)} \rightarrow \mathbb{R}$ that expresses how dissimilar an example appears to be with respect to a bag (or multi-set) of examples, assuming they are all iid. A Non-Conformity Measure can be in principle extracted from any Machine Learning (ML) algorithm. Although there is no universal method to derive it, a default choice is:

$$A((x, y), \wr z_1, \ldots, z_k \int) := -\Delta(y, f(x))$$

where $f : \mathbf{X} \rightarrow \mathbf{Y}'$ is the prediction rule learned on $(z_1, \ldots, z_k)$ and $\Delta : \mathbf{Y} \times \mathbf{Y}' \rightarrow \mathbb{R}$ is a measure of similarity between a label and a prediction.

Armed with an NCM, it is possible to compute for any example (x, y) a p value that has the following property: for any chosen $\epsilon \in [0, 1]$, the p value of test examples (x, y) drawn iid from the same distribution as the training examples are (in the long run) smaller than ϵ with probability at most ϵ.

The idea is then to compute for a test object a p value of every possible choice of the label. Once the p values are computed, they can be put to use in one of the following ways:

- Given a significance level ϵ, a *region predictor* outputs for each test object the set of labels (i.e., a region in the label space) such that the actual label is not in the set no more than a fraction ϵ of the times. This is called the *validity* property. It provides a long term guarantee on the number of errors (where "error" is defined as "actual label not in the prediction set") in the long run. If the prediction set consists of more than one label, the prediction is called *uncertain*, whereas if there are no labels in the prediction set, the prediction is *empty*.
- Alternatively, one can take a *forced* prediction (where the label with the largest p value is chosen for a given test object), alongside with its *credibility* (the largest p value) and *confidence* (the complement to 1 of the second largest p value).

There are two forms of CP: *Transductive CP* (TCP) and *Inductive CP* (ICP). They differ in the way in which the NCM (and consequently the p values) are calculated. TCP is computationally expensive as the computation of the NCM is performed from scratch for each example. Formally, in TCP

$$\alpha_i = A((x_i, y_i), \wr z_1, \ldots, z_\ell, z_{\ell+1} \int / (x_i, y_i)) \qquad i = 1, \ldots, \ell + 1$$

which means that for every α_i (there is one α_i for every training example plus the hypothetical example) the underlying algorithm is trained anew on a training set in which the example to

[1] in fact, even a weaker requirement of *exchangeability* is sufficient.

which the α_i is associated has been removed. A prediction is then obtained for the example on the model thus trained and the NCM is derived.

By contrast, Inductive CP requires just one training of the underlying ML algorithm. The overall training set is split into a proper training set and a calibration set. The proper training set is used to train the underlying ML algorithm. The α_i are computed only on the examples of the calibration set.[2] Assuming that the first m examples constitute the calibration set and the remaining $k = \ell - m$ examples the proper training set, the α_i can be formally expressed as:

$$\alpha_i = A((x_i, y_i), \wr z_1, \ldots, z_k \int /(x_i, y_i)) \qquad i = 1, \ldots, m + 1$$

Once the NCM have been calculated and denoting with n their number (which is $\ell + 1$ for TCP and $m + 1$ for ICP), the p value for a hypothesis $y_{\ell+1} = \bar{y}$ about the label of test object $x_{\ell+1}$ is defined as follows:

$$p_{\bar{y}} = \frac{|\{i = 1, \ldots, n : \alpha_i \geq \alpha_{\ell+1}\}|}{n}$$

Both the Transductive form and the Inductive form of CP are proven to have the validity property when the prediction region Γ_ϵ for a test object x for a chosen significance level $\epsilon \in [0, 1]$ is the set of labels for which the p value exceeds the significance level:

$$\Gamma_\epsilon(x) := \{y \mid p_y > \epsilon\}$$

Finally, the validity property as stated above guarantees an error rate over all possible label values, not on per-label value basis. The latter can be achieved with a variant of CP, called *label-conditional CP* (a variant of Mondrian CP). The only change is in the calculation of the p value: we restrict the α_i only to those that are associated with examples with the same label as the hypothetical label that we are assigning at the test object. So, the p value for a hypothesis $y_{\ell+1} = \bar{y}$ about the label of test object $x_{\ell+1}$ is defined as follows:

$$p(\bar{y}) = \frac{|\{i = 1, \ldots, (\ell + 1) : y_i = \bar{y}, \alpha_i \geq \alpha_{\ell+1}\}|}{|\{i = 1, \ldots, (\ell + 1) : y_i = \bar{y}\}|}$$

The property of label-conditional validity is essential in practice when the CP is applied to an “imbalanced” data set, i.e. a data set in which the proportions of labels are significantly different. Empirically, one can observe that with the plain validity property, the overall error rates tend within statistical fluctuation to the chosen significance level, but the minority class(es) are disproportionally affected by errors. This property ensures that, even for the minority class, the long-term error rate will tend to the chosen significance level.

3 Combination

The study of the problem of combining p values to obtain a single test for a common hypothesis has a long history, originating very soon after the framework of statistical hypothesis testing was established (Fisher 1932). A survey can be found in Loughin (2004). In its more general form, the problem raised a lot of attention for its application to meta-analysis, where the results of a number independent studies, generally with different sample sizes and different procedures, are combined. The various methods that have been proposed over the years

[2] It could be argued that Inductive CPs require more data because they need a calibration set in addition to a proper training set. Cross-conformal predictors (Vovk 2015) can mitigate this issue, but their validity is weaker.

have tried to cater for the different ways in which the evidence manifests itself. In particular, some methods allow for weighting, thereby assigning more importance to some p values (for instance, in the case of meta-analyses, those corresponding to studies with larger samples sizes). More importantly, each method is associated with a different shape of the rejection region (the portion of the k-dimensional space of the k p values being combined for which the combined test of significance would reject the null hypothesis under a chosen significance level ϵ). The shape reflects the different way in which evidence of different strength is incorporated into the aggregated p value. It has been observed that there is no single combination method that outperforms all others in all applications.

The combination of p values from different Conformal predictors on the same test object is a very special form of the general problem outline above.

A method for the combination of Conformal Predictors should aim to:

- *Preserve validity* for the output of the combination method to be a valid Conformal Predictor, this is a necessary property.
- *Improve efficiency* smaller prediction sets should result from a desirable method of combination.

In practice, one is interested in the two desiderata above if the resulting p values are to be used to obtain prediction sets. There are domains of application where the p values can be used in other ways. An example which will be developed further in the sequel is in the context of Drug Discovery: the p values can be used to rank candidate compounds (see Toccaceli et al. 2017) in terms of the confidence in their activity (or lack of confidence in their inactivity), so that an informed decision can be made as to which candidate compounds to choose for a new batch of screenings.

There are two key observations that apply to p values computed by Mondrian Inductive Conformal Predictors (MICP):

1. *The p values from the same Conformal Predictor for the various test objects do not necessarily follow the uniform distribution.* The p values in Statistical Hypothesis Testing are uniformly distributed by construction if the null hypothesis is true. Similarly, when one examines the MICP p values for a set of test objects, it is apparent that only those for which the hypothetical label assignment is the correct one are uniformly distributed. The p values for the objects for which the hypothetical label assignment is incorrect tend to have values close to 0.
2. *The p values from different Conformal Predictors for the same test object are not independent.* One has to expect that, when testing the same hypothesis with different methods on the same object, the results will exhibit some degree of correlation. In other applications of p value combination, the issue may be less of a concern. For instance, in meta-analyses of clinical trials, it is arguable that there is less correlation because the trials are not reusing the same patients in the same groups. However, the one considered is certainly not the only context in which dependent p values are encountered and the issue has attracted some attention by statisticians (Pesarin 2001; Brown 1975; Alves and Yu 2014; Poole et al. 2016).

3.1 Methods from "traditional" statistical hypothesis testing

As outlined in Loughin (2004), the field of Statistical Hypothesis Testing has approached the problem of p value combination as soon as the notion of p value started to establish itself in the statistical community. One can identify, broadly speaking, two classes of p value combination methods: quantile methods and order-statistic methods.

Order-statistic methods (Davidov 2011) are mentioned here for completeness. Given k p values coming from k experiments, the combining function is based on the order of the p values. For instance, a combination method might simply consist in taking the smallest of the p values; another method might take the second smallest, another the arithmetic average, another the maximum and so forth. Intuitively, a combination method that takes the smallest p value "believes" the outcome that is most improbable under the Null Hypothesis, whereas a methods that takes the largest p value would be stricter, in that it would take the outcome that is less contrary to the Null Hypothesis. In general, such methods appear to be inadequate for Conformal Predictors, because they result in the loss of validity.[3]

On the other hand, quantile methods can satisfy this requirement. The quantile methods can be generally constructed by transforming the p values using a function chosen as the inverse of a Cumulative Distribution Function (CDF) of a convenient distribution. The choice of the distribution is in principle arbitrary, but it is convenient to constrain it to those distributions for which the CDF of the sampling distribution of the sum of Random Variables (RVs) can be expressed with closed formulas or can be calculated with little computational effort (it has been noted (Zaykin et al. 2007) that "nowadays any form of CDF can be used with the aid of simple Monte Carlo evaluation"). Let's assume that $\mathbb{D}$ is a distribution with support $[a, b]$ and with invertible CDF $F_X(t) : [a, b] \to [0, 1]$. A quantile method would transform the p values p_i (now considered Random Variables) into Random Variables $T_i = F_X^{-1}(p_i)$. By construction, these T_i are distributed according to $\mathbb{D}$. If we call $F_{T_1+T_2+\cdots+T_k}(t)$ the CDF of the sum of k $\mathbb{D}$-distributed RVs, the combined p value is obtained as $p_{\text{comb}} = F_{T_1+T_2+\cdots+T_k}(t_1 + t_2 + \cdots + t_k)$. It is easy to see that P_{comb} is uniformly distributed, based on elementary property that will be proved in a later section.

Here we consider one quantile method, namely Fisher's method (also known as chi-square method), although other quantile methods exist, such Stouffer's method (1949) (also known as z-transform test).

3.2 Fisher's method

Fisher's method (1932; 1948) is among the earliest p value combination methods. It relies on the key observation that if $p_1, p_2, \ldots, p_k$ are each the realization of a uniformly distributed random variable,

$$h_i = -2 \log p_i \qquad \text{with} \quad i = 1, \ldots, k$$

is a random variable that follows a chi-squared distribution with 2 degrees of freedom.

The sum of k independent random variables each following a chi-squared distribution with 2 degrees of freedom is itself chi-squared distributed with $2k$ degrees of freedom, that is:

$$h = -2 \sum_{i=1}^{k} \log p_i$$

is a random variable that follows a chi-squared distribution with $2k$.

The combined p value is:

$$p = \mathbb{P}\left\{ y \leq -2 \sum_{i=1}^{k} \log p_i \right\}$$

where y is a random variable following a chi-square distribution with $2k$ d.f. The integral required for calculating the probability above has a very simple closed form:

[3] Note that validity is preserved when using the combination method that takes the smallest of the p values.

$$t \sum_{i=0}^{k-1} \frac{(-\log t)^i}{i!}$$

where $t = (p_1 \times p_1 \times \cdots \times p_k)$.

Interestingly, the formula above also arises as the probability of the product of independent uniform random variables (Zaykin et al. 2002). Fisher's method also exhibits a form of asymptotic optimality "among essentially all methods of combining independent tests" (Littell and Folks 1973).

3.3 Validity recovery

None of the methods discussed so far guarantees validity. Fisher's method guarantees valid p values but only under the assumption of independence. The resulting combined p value will exhibit a deviation from the uniform distribution that will be more pronounced the stronger the dependence among p values. In our specific setting, the p values are obtained by applying CP with different underlying algorithms on the same test object. It is therefore to be expected that the p values will exhibit a substantial degree of correlation. Figure 1 illustrates the effect of correlation on combination.

The problem is well-known and there have been many attempts to introduce corrections based on some measure of the correlation or of the dependence. We propose a very simple calibration method, based on the following well-known elementary result.[4]

Given a random variable X and its CDF $F_X(t) \equiv \mathbb{P}[X \leq t]$ which we'll assume invertible (but slightly less restrictive assumptions are possible), the random variable $Y \equiv F_X(X)$ follows the Uniform distribution.

This can be proved by showing that the CDF of Y is the identity function, i.e. $F_Y(y) = y$, along the following lines:

$$F_Y(y) \equiv \mathbb{P}[Y \leq y] = \mathbb{P}[F_X(X) \leq y] = \mathbb{P}\left[X \leq F_X^{-1}(y)\right] = F_X\left(F_X^{-1}(y)\right) = y$$

The method we propose consists in calibrating the combined p value obtained with any of the methods above by using the CDF of the combined p values. An estimate of such CDF can be obtained from the Empirical Cumulative Distribution Function (ECDF) observed on a Calibration Set (any set drawn from the same distribution, with the exclusion of the training set and the test set).

In the interest of clarity, we state the steps involved ECDF-based calibration in Algorithm 1

```
Data:
  $p$ values $p_{\mathrm{cal},i}$ from a calibration set,
  $p$ values $p_{\mathrm{raw},j}$
Result: calibrated $p$ values $\hat{p}_j$
Compute Empirical Cumulative Distribution Function $F_X(x)$ of values $p_{\mathrm{cal},i}$ in calibration set;
Apply $F_X()$ on the values to calibrate, obtaining calibrated $p$ values $\hat{p}_j = F_X(p_{\mathrm{raw},j})$;
```

Algorithm 1: ECDF-based p value calibration

[4] A reviewer pointed out that a similar approach has already been described in Balasubramanian et al. (2015) and empirically evaluated,, with mixed results, by Linusson et al. (2017).

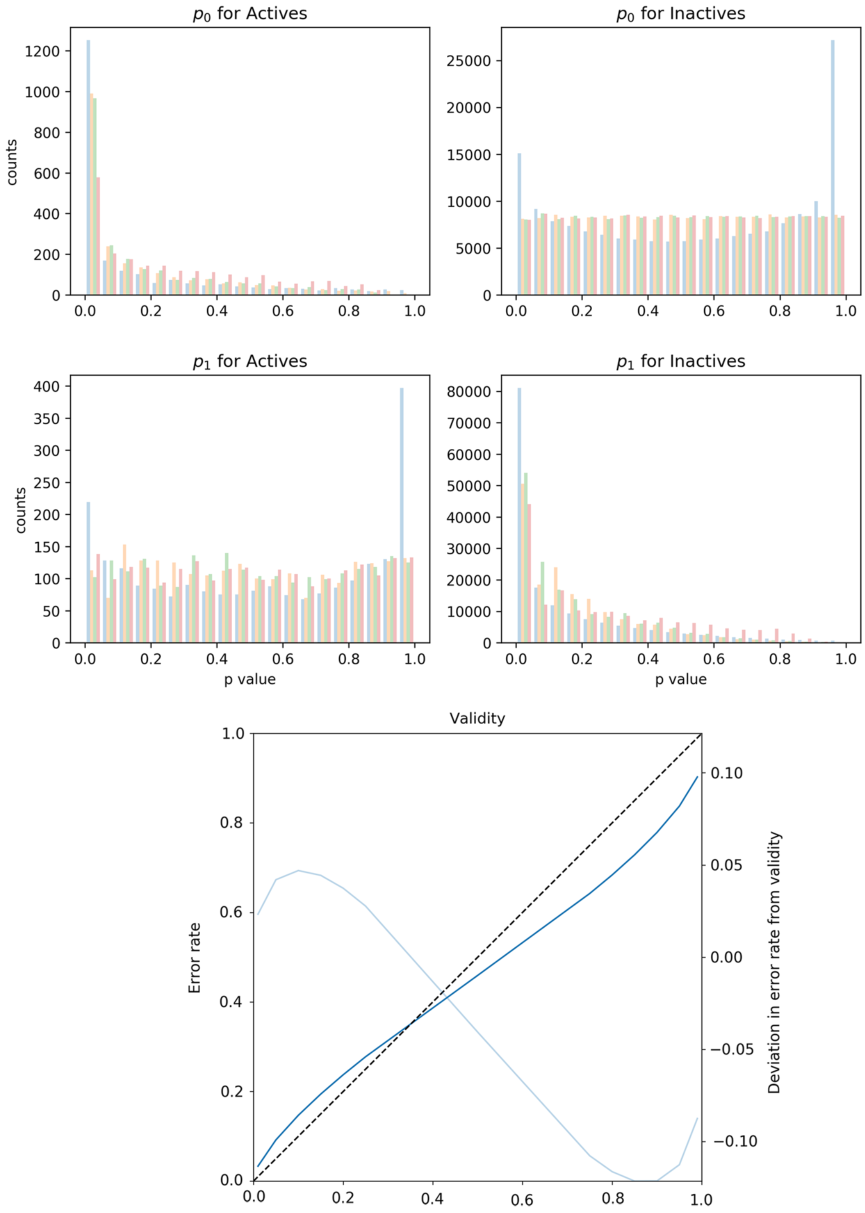

Fig. 1 Deviation from validity when combining with Fisher's method. The four histograms at the top show the counts of p values over 20 equally sized bins between 0 and 1 from one of the runs of real-world example discussed later. In each bin, the bars refer to the combined CP (the leftmost bar, in light blue) and three base CPs. The p values $p_{\bar{y}}$ of test objects with label $\bar{y}$ (i.e. same label as the one for which we are computing the p value) should be uniformly distributed, whereas they should be concentrated towards 0 for test examples with other labels. The uniform distribution is essential for the validity property. We should observe a uniform distribution for the upper right and lower left histograms. The correlation between p values for the same object caused Fisher's method to deviate from uniformity. The Pearson correlation ranged between 0.44 and 0.77 for p_0 and between 0.36 and 0.78 for p_1. The effects on validity are shown in the bottom plot. The blue line should be overlapping the dashed line. The light blue line (whose y-axis is on the right) shows the difference between blue line and the dashed line

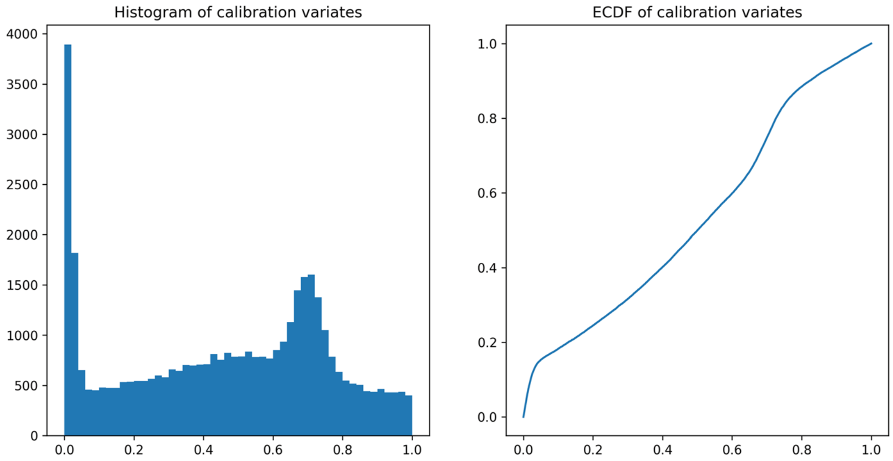

Fig. 2 On the left, the plot is the histogram of about 40,000 variates of an arbitrary illustrative distribution $q(x)$ with values between 0 and 1. On the right, the plot shows the Empirical Cumulative Distribution Function $ECDF(x)$ obtained on the basis of the variates

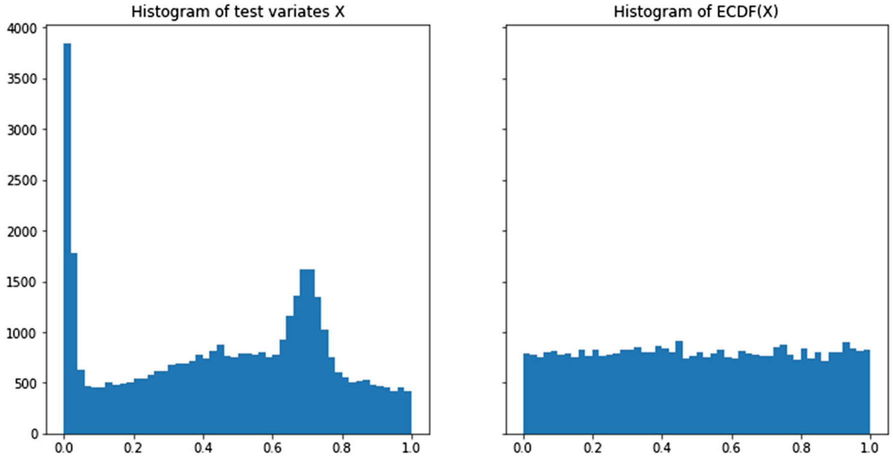

Fig. 3 On the left, the plot shows the histogram of about 40,000 variates drawn from the same distribution $q(x)$ shown already in Fig. 2. The right plot shows the histogram of the values $y = ECDF(x)$ obtained by applying the function $ECDF(x)$ shown in Fig. 2 on the $\approx 40,000$ variates of the left plot

It is worth pointing out that, when calibrating $p_{\bar{y}}$, the calibration set should contain the p values of the calibration examples with label $\bar{y}$, because these p values are supposed to be uniformly distributed.

Figures 2 and 3 illustrate how the ECDF-based calibration operates on variates from an example distribution.

3.4 Learning to combine

In the methods discussed so far, the combined p value is a function only of the p values from the individual CPs. The combination function does not take into account the object to which the p value refers. Intuitively, it seems legitimate to wonder whether there are gains to be made by making the combination a function also of the object. Indeed, it may be argued that different underlying algorithms, especially when they are intrinsically different, might exhibit differences in relative performance on different objects: algorithm 1 might perform generally better than algorithms 2 and 3 in a certain region of the object space, whereas algorithm 2 might be better than the others in another region, and so on. The combination function could be learned by means of an appropriate ML method. Although the idea of learning the combination function is not new (see Balasubramanian et al. 2015), we believe that the approach we propose is novel.

Before setting up the learning problem, it may be useful to discuss what the ideal p value combination should look like. Ultimately, the objective is to obtain a CP that outputs p values for class c that are:

(a) uniformly distributed for objects belonging to the c class
(b) 0 for objects belonging to other classes.

In any practical application, the latter objective is really to obtain p values as close to 0 as possible for objects belonging to the other classes. A CP that outputs such p values would exhibit validity and maximum efficiency (where we define efficiency as the average size of the region prediction).

With these objectives in mind, we can formulate the problem of CP combination as the problem of predicting, given an object, which of the k base CPs will provide the p value that best approximate requirements (a) and (b). In fact, a soft version of this formulation (as opposed to the 'hard' decision) might seek the weights with which to combine the individual CP predictions to best approximate requirements (a) and (b).

One possible setting of the problem is to use Logistic Regression with a training set constructed in the following manner. Suppose we have a set of objects $\mathbf{x}_i$, labels y_i, and p values $p_{\text{act},i,j}$ and $p_{\text{inact},i,j}$. We'll refer to this as combination training set. We use this to create a combiner training set, which is specific to the class c for which we are creating a combiner. Let's consider the combiner for p_{Active}. For every example $(\mathbf{x}_i, y_i)$, the combination training set has k examples $(\mathbf{x}_i, 1), \ldots, (\mathbf{x}_i, k)$, i.e. one for each of the k base CPs. Each of these k examples is assigned a weight $w_{i,j}$, whose value is calculated as a function of the label y_i and of p value $p_{\text{act},i}$. We considered 2 different ways, which are discussed next, but many more schemes are possible.

To illustrate how the method works, Fig. 4 shows an example of its application to a synthetic data set. In this admittedly contrived data set, the LR-based combiner manages to assign probabilities to each CPs that result in a combined CP with the best properties of each base CP (compare Fig. 4c, d with Fig. 4f.

3.4.1 Method 1: weighted

In this method the weight is higher for lower p values when the example is not active and higher for higher p values when the example is active (in the case of setting value of the p_{inactive} combiner, it would be the other way round).

$$w_{i,j} = \begin{cases} \frac{1-p_{\text{act},i,j}}{\sum_{j=1}^{k}(1-p_{\text{act},i,j})} & \text{when } y = \text{inact} \\ \frac{p_i}{\sum_{j=1}^{k} p_{\text{act},i,j}} & \text{when } y = \text{act} \end{cases} \tag{1}$$

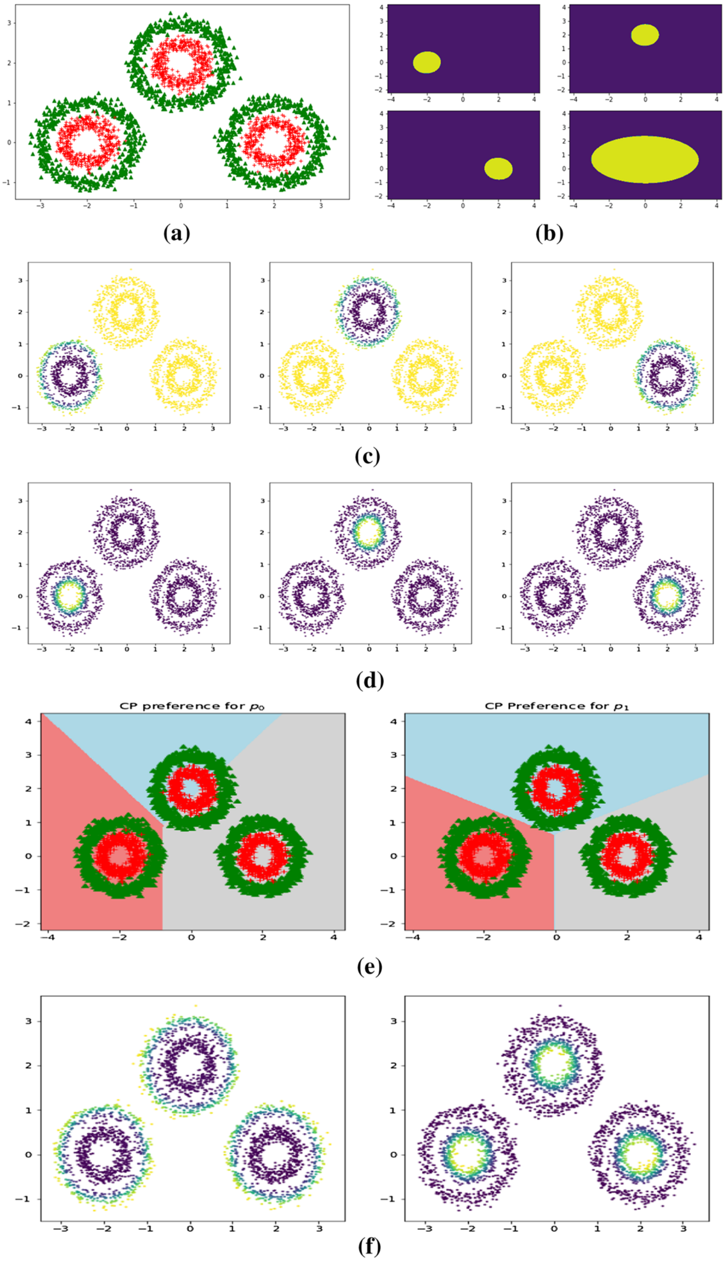

Fig. 4 **a** The synthetic data set, made up of three pairs of concentric circles, the inner containing positive examples and the outer negative examples. Three separate CPs are obtained by using three separate SVCs (polynomial kernel of degree 2) as underlying algorithms. Each SVC is trained on only one of the three clusters. The decision function for each of the SVCs is plotted in b, with yellow denoting positive values and purple negative values. The lower bottom panel in **b** refers to an SVC that was trained on the entire data set. **c** and **d** p_0 and p_1 from each of three CPs on a test set from the same distribution as the overall training set. Purple corresponds to low values, green to intermediate, and yellow to high values. In **e** the color (coral, light blue, grey) corresponds to the CP with largest probability predicted by the LR combiner in that point. Finally, in **f** the values of p_0 and p_1 shown with the same color coding as in **c** and **d**

The weights are also normalized so that the weights for the same object add to 1. Also, per-class weighting is applied to compensate for imbalance. Specifically, the examples of the Active class are weighted by $N_{Inactive}/(N_{Inactive} + N_{Active})$ and those of the Inactive class by $N_{Active}/(N_{Inactive} + N_{Active})$.

Note that the combiner training set is k times as big as the combination training set.

One issue with this method is that, while it may seem desirable to favour base CPs that produce higher p values for Active examples, this does not appear to go in the direction of meeting requirement (a).

3.4.2 Method 2: reduced

The 'reduced' method addresses the potential issue mentioned at the end of the previous paragraph by simply not including examples in the combiner training set if they have the Active (correct) label. In this way, we do not induce the combiner to favour higher p values for Active examples.

$$w_{i,j} = \begin{cases} \frac{1-p_{\mathrm{act},i,j}}{\sum_{j=1}^{k}(1-p_{\mathrm{act},i,j})} & \text{when } y = \text{inact} \\ 0 & \text{when } y = \text{act} \end{cases} \tag{2}$$

The examples that are assigned weight 0 can be discarded from the combiner training set. This is quite advantageous as it reduces the size of the training set, which is otherwise k as big as the combination set.

3.4.3 Predictions: hard and soft variants

As hinted in Sect. 3.4, the methods above produce a value that can be interpreted as a weight assigned to each base CP, which can be then used to compute the predicted p value for a test object in two ways: (a) by *hard* choice, i.e. by taking the p value of the base CP with the largest weight, (b) by *soft* combining, i.e. by computing a linear combination of the base CP p values using the predicted weights.

4 An application: ExCAPE

In the sequel, we are going to present some results of the application of the combination methods to a real-world problem, which the authors encountered during their participation to a EU project called ExCAPE (Exascale for Compound Activity Prediction Engines) for the prediction of the activity of chemical compounds towards biological targets of interest.

Any advances in the ability to predict correctly such activity are of extreme interest to the pharmaceutical industry, as this can reduce substantially the number of lab assays required to identify new active compounds, thereby resulting directly in lower costs and a competitive advantage in the ability to innovation. While one could argue that computer-aided drug discovery has been around for at least 30 years, the "Exascale" qualification in the name of the project alludes to one distinguishing feature of this research effort. ExCAPE explores methods that can be parallelized extensively, towards the goal of exploiting efficiently Exascale High Performance Computing (HPC) platforms, i.e. computing platforms capable of an aggregated 10^{18} FLOPS (Floating Point Operations per Second). This level of scalability is rapidly

Table 1 Key statistics of the IDH1 data set. The lower part refers to the data sets used in each of the 50 runs

Total number of examples = 468,798
Number of features = 639,253
Number of non-zero entries = 31,523,836
Density of the data set = 0.0001
Active compounds = 2,194 (1.3%)
Proper training set size = 200,000
Calibration set size =100,000
Test objects = 168,798

becoming relevant as it is expected that Exascale systems will become available in the 2020 timeframe.[5]

4.1 Multiple compound activity predictions

The specific problem tackled by ExCAPE is to predict the activity of a large number of compounds (several hundred thousands of compounds) towards a number of targets of interest (less than a thousand). The biological activity is known only for a fraction of the compound-target combinations. The challenge is to predict the activity in the large proportion of unknown compound-target cases. Different ML approaches are being pursued concurrently and separate heterogeneous models are being developed, namely Multi-task Deep Neural Networks and Bayesian Matrix Factorization. This created the need for a way to combine the predictions of these models (and possibly others) into one final set of predictions. Conformal Predictors can address this need by offering a solid framework for calibrating predictions expressed in different scales into p values and then enabling their advantageous combination with the techniques described in this paper.

4.2 Chemoinformatics data sets

The data set used in the experiments in this paper was extracted from the latest version of the reference data base of the project, called ExCAPEDB. The specific target, identified as IDH1, which was chosen because it's the one with the largest number of tested compounds. The characteristics of the data set are reported in Table 1. The compounds are represented by means of features that capture structural properties of the molecule. From the standpoint of Machine Learning, chemoinformatics data sets can present the following challenges:

- Sparsity: the design matrix (the matrix that has a column for each feature and a row for each object) is often extremely sparse. In the case of the IDH1 data set, the fraction of non-zero entries is 0.01
- Imbalance: the prevalence of active compounds is often very small, 1% or less.

The three algorithms selected for this investigation are: linear SVC, Gradient Boosted Trees, and k Nearest Neighbours. The choice was driven by the intuition that ML algorithms based on inherently different approaches might have complementary strengths and weaknesses that a combination method could exploit to its advantage. Other choices might have

[5] Just for reference, at the time of writing, the `top500.org` site reports that the most powerful supercomputer is the Sunway TaihuLight at the National Supercomputing Center in Wuxi, China, rated at $\approx 0.093 \times 10^{18}$ FLOPS, i.e. about 1/11 of what would be considered Exascale.

been equally valid: for examples, the same kernel method with different kernels, or the same regularized algorithm with different values of the regularization parameter, or distance-based algorithms with different types of distances.

5 Results and discussion

5.1 Experimental setup

The experiments were primarily run on the computing facilities offered by the IT4Innovation National Supercomputing Center, Ostrava, Czech Republic. The Center operates two clusters, Anselm and Salomon, with 209 nodes and 1008 nodes respectively. The nodes are powerful servers, each equipped in case of Anselm with 16 cores (2 Intel Sandy Bridge E5-2665, 2.4 GHz) and 64 GB of RAM, and in the case of Salomon with 24 cores (2 Intel Xeon E5-2680v3, 2.5 GHz) and 132 GB of RAM.

The software was developed in Python, in large part using Jupyter Notebooks (Kluyver et al. 2016). The `scikit-learn` (Varoquaux et al. 2015) package provided implementations for linear SVC, Gradient Boosted Trees and k Nearest Neighbours, whereas the preparation and handling of the data and of the results was carried out using the `numpy` (van der Walt et al. 2011), `scipy` (Jones et al. 2001), and `pandas` (McKinney 2010) packages. The distribution of the computation over the nodes obtained for a job relied on the `distributed` (Dask Development Team 2016) package, which allows Python functions (or more in general Directed Acyclic Graphs (DAGs) of Python functions) to be submitted to a central "scheduler" for execution on a distributed cluster of workers.

Guaranteeing a full utilization of the nodes proved less straightforward than anticipated. Despite the computation consisting of fundamentally independent runs (a case of what is referred to as 'embarrassing parallelism'), it turned out that different algorithms had different CPU usage profiles and memory usage profiles, so the parameters governing the distribution (e.g. number of workers per node, maximum number of outstanding remote calls) had to be carefully tuned to avoid bottlenecks or memory overloads, especially on the node hosting the scheduler. The practical difficulty was compounded by the unexpected level of congestion on the Salomon and Anselm clusters, which meant that the number of nodes requested often had to be scaled back (all the way down to 6 or 8) to have a chance to be allowed to run.

The execution times for the LR-based Combination were dominated by the training times for the higher values of the regularization parameter C. For values larger than 1000, training time would be in the order of tens of minutes, whereas it would be of order of seconds for small values of C (heavy regularization). On a 16-node cluster on Anselm, one 10-fold CV over a range of 25 logarithmically-space values from 10^{-6} to 10^{6} required ≈ 1 h.

5.2 Results

The original IDH1 data set was used to obtain 50 partitions into training, calibration, and testing sets. The training set size was chosen as 200,000 and the calibration set size as 100,000, leaving 168,798 examples for the test set. The splits were stratified, i.e. each set has the same proportion of the two classes as in the overall set. Linear SVC, Gradient Boosted Trees, and k Nearest Neighbour models were trained for each of the 50 splits and scores were obtained for the calibration and testing sets. The choice of the specific algorithms was driven by the intuition that algorithms based on fundamentally different concepts would have different

Table 2 The Non conformity measures for the three underlying algorithms

Underlying	Non conformity measure α_i	Comment
SVM	$-y_i f(x_i)$	where $f(x_i)$ is the SVM decision function
kNN	$\dfrac{\sum_{j \neq i: y_j = y_i}^{(k)} d(x_j, x_i)}{\sum_{j \neq i: y_j \neq y_i}^{(k)} d(x_j, x_i)}$	where $d(x_i, x_j)$ is the (Euclidean) distance; the summation is on the k smallest values of $d(x_j, x_i)$
XGB	$-y_i p(y_i = +1 \vert x_i)$	$p(y_i = +1 \vert x_i)$ is the probability of Activity estimated by the classifier

strengths and weaknesses in different regions of the problem space and would have a higher chance of complementing each other. Parameter optimization for was performed once for each of the algorithms. In all cases the reference metric was the F1 score. The potentially suboptimal performance deriving from a single setting of the parameters was not deemed to be a problem: the focus of the investigation is indeed on the combination of the Conformal Predictors, rather than on their individual quality. In fact, the variability of performance across splits might add a useful element of diversity in the relative merit of the predictors.

After obtaining p values via Inductive Mondrian (Class-conditional) Conformal Predictors for each underlying algorithm using the NCM detailed in Table 2, we turned to the combination of p values.

The application of the passive methods was obviously straightforward, whereas the LR-based combination required the majority of effort. For each of the runs, the LR classifier was obtained with a parameter optimization via 10-fold cross validation over 13 logarithmically-spaced values from 10^{-4} to 10^{2}. The calibration set (i.e. the set used for the Inductive CP) was also used as combination training set, from which the two combiner training sets (one for the combination of p_{active} and one for the combination of p_{inactive}) were derived. Note that, as explained in Sect. 3.4, the combiner training sets are $k = 3$ times as large the combination set, hence their size is as large as 300,000.

The performance of the combined CP is examined on confusion matrices and ranking. The confusion matrix for CP region prediction is slightly different from the usual one for traditional classifiers as, in addition to a breakdown into correct and incorrect precise predictions, it includes a count of the empty predictions and a count of the uncertain predictions (the uncertain predictions occur when the region predictor contains more than one label). The metrics for CP confusion matrices include Precision and Recall. For reference, the definitions used in this study are summarized in Table 3. In order to have just one metric, we combine Precision and Recall into the F_1 score, which is their harmonic mean and is a special case of the F_β score, where β controls the "preference" of Precision vs. Recall.

The ranking performance is evaluated in terms of precision-at-k and average precision. The latter provides an overall view of how high in the ranking the examples belonging to the Positive class (here, the Active compounds) were placed. The precision-at-k offers an assessment of the ranking that is more focused on the top, which in many applications is what matters most. Precision-at-k is simply the fraction of Positive labels in the top k objects in the ranking. If, for instance, in a drug discovery setting only the top k compounds are chosen for actual lab testing, then it is arguable that average precision is not relevant and we want to select the method that places the largest fraction of Actives in the top k. However, the situation might be different if the intended use is for test prioritization, in which case average precision might be relevant.

Table 3 Performance metrics used in this study

Metric	Definition	Comment
Precision	$Pr = \dfrac{\lvert PP \cap TP \rvert}{\lvert PP \rvert}$	Fraction of Positives among objects predicted as Positive
Recall	$Re = \dfrac{\lvert PP \cap TP \rvert}{\lvert TP \rvert}$	Fraction of all Positives included in the objects predicted as Positive
F_1 score	$F_1 = 2\dfrac{Pr \cdot Re}{Pr + Re}$	Harmonic mean of Precision and Recall
Precision-at-k	$Pr@k = \dfrac{\lvert PP_k \cap TP \rvert}{\lvert PP_k \rvert}$	Fraction of Positives in the top k ranked objects
Average precision	$AP = \dfrac{1}{m}\sum_{j=1}^{m} Pr@k_j$	Average over each Positive of the precision-at-k_j where k_j is its position in the ranking

TP True positive, PP predicted positive, PP_k predicted positive within the k top ranked objects, m number of Positives, k_j ranking position of j-th Positive

Finally, statistical significance of the results is estimated.We use a non-parametric statistical test on paired observations, namely the Wilcoxon signed-rank test (Wilcoxon 1945; Hollander and Wolfe 1999). The null hypothesis of the Wilcoxon signed-rank test is that the distribution of the differences between elements of pairs is symmetrical around 0. However, in its basic form, the test does not apply to variables with discrete values such as counts but only to variables with continuous values, the reason being that the test was not designed to deal (*a*) with no differences in a pair and (*b*) with ties among the differences (occurrences of pairs with the same difference in absolute value). Variants have been proposed [by Wilcoxon himself, who suggested to disregard the observation pairs with no difference, and in Pratt (1959), who suggested a way to account for those] but the distribution of the statistic would change.

5.3 Region predictions

For the sake of brevity, we show only the counts for significance level $\epsilon = 0.01$ (Table 4) and we refer the reader to the Supplementary Material for the confusion matrices for other significance levels. We also report the error rate to provide a view on the validity deviation. If one compares error rates with and without it, ECDF-based calibration can be seen recovering validity very effectively.

The overall view of the strengths and weaknesses of the various methods across different significance levels is captured in Table 5, where we show the values of the F_1 scores for the Active class as well as for the Inactive class. The interpretation of the results is not straightforward. In the case of the Active class, there is no single method that outperforms consistently all the others in terms of the F_1 score. It is debatable whether non-valid methods should be considered, but they were reported to let the reader get a sense of how the ECDF-based calibration affects te performance. Among the simpler valid methods, "Fisher ECDF" improves over any base CP. The statistical significance the Wilcoxon test attributes is at least at the level of $p < 1.3 \times 10^{-5}$. On the other hand, the LR-based combination methods fail to improve over the base CPs, especially in their valid variant.

In the case of the Inactive class, however, the "Weighted Hard ECDF-calibrated" method exhibits very good performance, with also "Weighted Soft ECDF-calibrated" scoring very

Table 4 Confusion matrices for significance levels $\epsilon = 0.01$ (4a) and $\epsilon = 0.05$ (4b)

Method	ApA	ApI	IpI	IpA	Empty	Uncertain	Error rate
(a)							
SVC	598.06	22.58	17,937.00	1648.08	0.00	148,592.28	0.010
XGB	570.70	21.66	27,929.04	1650.92	0.00	138,625.68	0.010
kNN	339.12	20.64	24,188.68	1666.54	0.00	142,583.02	0.010
Min	774.96	53.20	37,467.94	3709.34	31.96	126,760.60	0.022
Max	217.02	1.52	9725.14	197.46	0.00	158,656.86	0.001
Mean	275.26	2.70	14459.18	314.64	0.00	153,746.22	0.002
Fisher	941.06	83.00	50,909.08	5542.32	0.06	111,322.48	0.033
Min ECDF	575.94	24.54	28,212.10	1650.78	5.72	138,328.92	0.010
Max ECDF	468.72	20.98	27,002.12	1645.46	0.00	139,660.72	0.010
Mean ECDF	515.78	21.38	28,742.84	1656.98	0.00	137,861.02	0.010
Fisher ECDF	626.96	22.86	30,613.02	1655.20	0.00	135,879.96	0.010
Weighted soft	271.76	2.68	14,572.52	306.80	0.00	153,644.24	0.002
Weighted hard	425.08	35.18	30,855.70	874.06	0.66	136,607.32	0.005
Reduced soft	277.80	2.76	14,587.00	317.54	0.00	153,612.90	0.002
Reduced hard	610.14	40.16	32,519.74	2118.96	10.60	133,498.40	0.013
Weighted soft ECDF	524.30	37.86	35,798.94	1753.50	0.00	130,683.40	0.011
Weighted hard ECDF	586.96	244.88	79,351.32	2219.30	16.50	86379.04	0.015
Reduced soft ECDF	525.06	21.70	28,889.20	1658.02	0.00	137,704.02	0.010
Reduced hard ECDF	553.22	24.12	27,090.12	1654.58	4.20	139,471.76	0.010

Table 4 continued

Method	ApA	ApI	IpI	IpA	Empty	Uncertain	Error rate
(b)							
SVC	1010.18	113.14	49,788.14	8289.18	0.00	109,597.36	0.050
XGB	992.04	109.82	55,721.40	8286.66	0.00	103,688.08	0.050
kNN	565.78	113.52	41,779.04	8331.98	0.00	118,007.68	0.050
Min	1241.68	227.66	78,018.16	16,985.82	1650.70	70,673.98	0.112
Max	376.12	11.04	21,675.18	840.58	0.00	145,895.08	0.005
Mean	531.12	27.92	31,459.82	1795.16	0.00	134,983.98	0.011
Fisher	1245.82	219.96	80,820.78	15,273.98	39.72	71,197.74	0.092
Min ECDF	998.04	110.14	52,696.74	8100.72	215.38	106,676.98	0.050
Max ECDF	808.86	109.64	54,665.34	8302.60	0.00	104,911.56	0.050
Mean ECDF	931.06	111.38	57,696.06	8304.92	0.00	101,754.58	0.050
Fisher ECDF	1054.82	114.66	59,150.94	8294.36	0.72	100,182.50	0.050
Weighted soft	514.94	29.44	32,349.68	1663.24	0.00	134,240.70	0.010
Weighted hard	694.66	172.30	66,184.62	3776.18	51.72	97918.52	0.024
Reduced soft	552.04	29.68	32,329.56	1879.60	0.00	134,007.12	0.011
Reduced hard	1033.40	182.10	68,065.44	10,130.12	693.76	88,693.18	0.065
Weighted soft ECDF	934.64	195.72	77,314.34	8692.52	0.00	81,660.78	0.053
Weighted hard ECDF	931.30	558.70	120,985.68	9018.64	688.02	36,615.66	0.061
Reduced soft ECDF	969.38	111.82	58,102.40	8304.52	0.00	101,309.88	0.050
Reduced hard ECDF	965.92	110.10	52,332.94	8057.56	256.86	107,074.62	0.050

ApA, ApI, IpI, IpA refer to precise predictions, i.e cases in which the region prediction contained only one label; Empty refers to cases in which the prediction set was empty (both label could not be rejected); Uncertain refer to cases in which the region prediction contained more than one label. The Error rate allows to check whether the validity property is met. The Errors are the sum of ApI, IpI, and Empty. The number of test examples was 168,798. The values are averages over 50 runs.

ApA Active predicted active, *ApI* Active predicted inactive, *IpI* Inactive predicted inactive, *IpA* Inactive predicted active

Table 5 F1 score for precise predictions for various significance levels (averages over 50 runs).

Epsilon	F_1 for the active class				F_1 for the inactive class			
	0.01	0.05	0.10	0.15	0.01	0.05	0.10	0.15
SVC	0.269	0.176	0.122	0.096	0.193	0.459	0.617	0.716
XGB	0.258	0.173	0.121	0.096	0.287	0.501	0.645	0.736
kNN	0.161	0.102	0.077	0.066	0.253	0.401	0.510	0.593
Min	0.232	0.122	0.091	0.085	0.367	0.637	0.745	0.771
Max	0.166	0.220	**0.209**	**0.177**	0.110	0.230	0.334	0.429
Mean	0.198	0.235	0.187	0.142	0.159	0.317	0.477	0.603
Fisher	0.217	0.133	0.102	0.087	0.468	0.653	0.742	0.793
min ECDF	0.261	0.177	0.128	0.106	0.289	0.480	0.613	0.693
Max ECDF	0.218	0.143	0.106	0.088	0.279	0.493	0.630	0.711
Mean ECDF	0.236	0.163	0.120	0.097	0.294	0.514	0.656	0.743
Fisher ECDF	**0.280**	0.183	0.127	0.100	0.310	0.523	0.658	0.743
Weighted soft	0.196	0.235	0.192	0.148	0.161	0.325	0.496	0.634
Weighted hard	0.240	0.212	0.169	0.140	0.312	0.567	0.712	0.794
Reduced soft	0.199	**0.239**	0.190	0.141	0.161	0.325	0.495	0.630
Reduced hard	0.248	0.155	0.114	0.099	0.326	0.578	0.710	0.773
Weighted soft ECDF	0.235	0.158	0.116	0.094	0.353	0.633	0.782	0.856
Weighted hard ECDF	0.237	0.155	0.126	0.134	**0.643**	**0.839**	**0.897**	**0.896**
Reduced soft ECDF	0.240	0.169	0.123	0.099	0.295	0.517	0.661	0.749
Reduced hard ECDF	0.251	0.172	0.125	0.104	0.279	0.477	0.610	0.692

The best values are highlighted in bold

high. The Wilcoxon test confirms the statistical significance: the hypothesis of no difference between "Weighted Hard ECDF-calibrated" and "Fisher" (the closest competitor among the simpler methods) is rejected at the level of $p < 8 \times 10^{-10}$.

This difference in performance of LR-based methods between the two classes requires further investigation. One possibility is that per-class weighting mentioned in Sect. 3.4.1 did not adequately compensate for the class imbalance.

5.4 Rankings

The second perspective under which we study the possible merits of CP combination is in terms of ranking. The test objects can be ranked according to their p values in search of those that are most likely to be Active. In particular, we rank compounds by lowest $p_{inactive}$, i.e. by strength of the evidence against the Inactive hypothesis. We also rank compounds by highest p_{active}, although this may appear not to be justifiable within the framework of Statistical Hypothesis Testing. In fact, this appears empirically to provide good results. One justification might be that by ranking by highest p value we are indeed ranking objects by how likely it would be to pick—from a set drawn from the same distribution as the training set and calibration set—an example that would be more contrary to the hypothesis of randomness.

Table 6 Ranking precision for Actives expressed in terms of precision-at-k, for $k = 10, 25, 50, 100, 200$ and in terms of Average Precision

CP type	Ranked by	$k = 10$	$k = 25$	$k = 50$	$k = 100$	$k = 200$	Avg prec
SVC	Lowest p_0	0.652	0.647	0.638	0.618	0.580	0.180
	Highest p_1	0.632	0.648	0.636	0.617	0.579	0.180
XGB	Lowest p_0	0.614	0.580	0.578	0.545	0.509	0.165
	Highest p_1	0.604	0.601	0.575	0.547	0.511	0.165
kNN	Lowest p_0	0.698	0.703	0.691	0.657	0.603	0.106
	Highest p_1	0.720	0.714	0.690	0.656	0.603	0.106
Min	Lowest p_0	0.644	0.653	0.652	0.623	0.583	0.177
	Highest p_1	0.754	0.749	0.719	0.687	0.627	0.168
Max	Lowest p_0	0.752	0.752	0.718	0.684	0.627	0.152
	Highest p_1	0.616	0.656	0.648	0.613	0.574	0.156
Mean	Lowest p_0	0.758	**0.759**	0.717	0.688	0.632	0.171
	Highest p_1	0.756	0.754	0.719	0.688	**0.636**	0.195
Fisher	Lowest p_0	0.754	0.746	0.719	0.685	**0.636**	**0.200**
	Highest p_1	0.764	0.756	0.718	0.688	0.635	0.189
Weighted soft	Lowest p_0	0.764	0.756	0.718	0.689	0.631	0.170
	Highest p_1	0.760	0.754	**0.722**	**0.692**	**0.636**	**0.200**
Weighted hard	Lowest p_0	0.694	0.699	0.675	0.636	0.587	0.165
	Highest p_1	0.656	0.688	0.678	0.647	0.605	0.180
Reduced soft	Lowest p_0	0.756	0.756	0.716	0.691	0.631	0.176
	Highest p_1	**0.768**	0.753	0.717	0.689	0.633	0.190
Reduced hard	Lowest p_0	0.650	0.646	0.626	0.599	0.558	0.166
	Highest p_1	0.710	0.714	0.681	0.645	0.595	0.165

Best values in each column are highlighted in bold

The comparison of the ranking quality of the various methods is reported in Table 6, where we provide precision-at-k (we chose $k = 10, 25, 50, 100, 200$) and average precision. Note that, since the ECDF-based calibration is a monotone mapping, it does not affect the ranking, so there is no need to have separate cases for it. We report ranking precision for the Actives but not for the Inactives. Given the high imbalance (98.7% of the examples are Inactive), all methods managed to achieve the maximum score of 1 for all the 5 levels of Precision-at-k when ranking for inactivity. The Average Precision was also very high, exceeding 0.995 in all cases.

For the more challenging task of ranking for activity, the results indicate that combination in general improves significantly the precision across the board, compared to the base CPs. As a side note, it is surprising to see the kNN CP, which appeared to perform worse than SVC and XGB CPs in region prediction, achieve markedly higher precision-at-k (although the advantage disappeared for Average Precision).

LR-based combination, in particular in its “soft” variant, appears to be on a par with the simpler methods. While a simple visual inspection of Table 6 might suggest a tiny advantage for the ”Weighted Soft, highest p_1” variant, the Wilcoxon test applied to the corresponding precision values for “Fisher, lowest p_0” and “weighted soft, highest p_1” reveals that any differences are of no statistical significance ($p >> 0.05$).

5.5 Considerations and future directions

The LR-based p value combination can be of benefit when the different base CPs exhibit different relative performance in regions of the object space that can be well separated by the combination classifier. It may be the case that the separation of the domains can be performed effectively with a function space of lower complexity than the ones that are required for the predictions themselves. In our example, the limited gains, if any, of the LR-based combination may be ascribed to a highly non-linear (hyper)surface of separation of the various domains, which the linear LR could not resolve. A future direction of research might be to incorporate non-linearity in the Classifier used for combination (for instance, with Kernel Logistic Regression). Another form of non-linearity to be experimented with is in the assignment of weights to the examples of the combiner training set, which Eqs. 1 and 2 set as linear function of the base CP p values.

A further line of enquiry might be in approaching combination of CPs as a learning-to-rank problem, for which there is already a large body of research given its commercially valuable applications in Information retrieval (e.g. search engines). The p value can in fact be interpreted as expressing a fractional rank (the p value is the fraction of calibration set examples that are less conform than than hypothetical test example, so one can view this as the rank by non-conformity). With this approach, the combination would occur at the level of the NCM α_i.

Finally, on perhaps a more speculative note, combination opens new opportunities to assemble ML algorithms into classifiers capable of more complex tasks. We can envisage, for instance, a number of underlying algorithms with differing learning abilities providing predictions to a combiner which assesses their relative performance in the various regions of the problem space and learns how to best combine the individual predictions in a particular area. Underlying algorithms might be altogether different as in the real-world case discussed here or they might just have different level of regularization (to adapt to regions with different levels of noise) or they could be just trained each on a separate cluster of the overall data set, as in the synthetic data example, where their combination resolves regions of the data sets that one algorithm of the same class could not separate.

6 Conclusions

We have discussed a method to learn to combine Conformal Predictors that aims at preserving validity while improving efficiency and the other metrics of interest. We have suggested a method to recover validity, provided that enough data is available that some can be set aside for a calibration set. We have applied p value combination methods from Statistical Hypothesis Testing as well as the proposed ML-based combination methods to a challenging real-world example, discussing their relative merits under the respects of region predictions and of ranking. An Inductive Mondrian Conformal Predictor method has been applied to a strongly imbalanced data set (1.3% Active vs. 98.7% Inactive) with a total of almost half a million examples and over half a million features. We showed that combination methods such as Fisher's provide a statistically significant improvement over the individual CPs. In terms of precise predictions, ML-based combination methods showed no advantage for the Active class, but brought about significant improvements for the Inactive class; in terms of ranking, they improved on the base CPs, but not on the simpler combination methods.

Acknowledgements This project (ExCAPE) has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671555. We are grateful for the help in conducting experiments to the Ministry of Education, Youth and Sports (Czech Republic) that supports the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center—LM2015070". We are grateful to AstraZeneca for their support for the collaborative project entitled: "Machine Learning for Chemical Synthesis" (R10911). We thank Lars Carlsson of Stena Line as well as Zhiyuan Luo, Vladimir Vovk, Ilia Nouretdinov for many valuable comments and discussions.

References

Alves, G., & Yu, Y.-K. (2014). Accuracy evaluation of the unified P-value from combining correlated P-values. *PLoS One*, *9*(3), e91225. https://doi.org/10.1371/journal.pone.0091225.

Balasubramanian, V., Gouripeddi, R., Panchanathan, S., Vermillion, J., Bhaskaran, A., & Siegel, R. (Sept 2009). Support vector machine based conformal predictors for risk of complications following a coronary drug eluting stent procedure. In *2009 36th Annual Computers in Cardiology Conference (CinC)*, (pp. 5–8).

Balasubramanian, V. N., Chakraborty, S., & Panchanathan, S. (2015). Conformal predictions for information fusion. *Annals of Mathematics and Artificial Intelligence*, *74*(1), 45–65. https://doi.org/10.1007/s10472-013-9392-4.

Brown, M. B. (1975). A method for combining non-independent, one-sided tests of significance (corr: V32 p955). *Biometrics*, *31*(4), 987–992. ISSN 0006341X, 15410420.

Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. http://dask.pydata.org/en/latest/cite.html. Accessed 11 Aug 2018.

Davidov, O. (2011). Combining p-values using order-based methods. *Computational Statistics & Data Analysis*, *55*(7), 2433–2444. https://doi.org/10.1016/j.csda.2011.01.024.

Fisher, R. A. (1932). *Statistical methods for research workers* (4th ed.). Edinburgh: Oliver & Boyd.

Fisher, R. A. (1948). Question 14: Combining independent tests of significance. *The American Statistician*, *2*(5), 30–30.

Gammerman, A., & Vovk, V. (2007). Hedging predictions in machine learning (with discussion). *The Computer Journal*, *50*(2), 151–163. https://doi.org/10.1093/comjnl/bxl065.

Hollander, M., & Wolfe, D. A. (1999). *Nonparametric statistical methods.*, Series in Probability and Statistics New York, NY: Wiley.

Ivina, O., Nouretdinov, I., & Gammerman, A. (2012). Valid predictions with confidence estimation in an air pollution problem. *Progress in Artificial Intelligence*, *1*(3), 235–243. https://doi.org/10.1007/s13748-012-0018-6. ISSN 2192-6360.

Ji, G. R., Dong, Z., Wang, D. F., Han, P., & Xu, D. P. (2008). Wind speed conformal prediction in wind farm based on algorithmic randomness theory. In *2008 International conference on machine learning and cybernetics*, (vol. 1, pp. 131–135). https://doi.org/10.1109/ICMLC.2008.4620392.

Jones, E., Oliphant, T., & Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/, [Online; accessed 2017-04-09].

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., et al. (2016). Jupyter notebooks—A publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). Amsterdam: IOS Press.

Lambrou, A., Papadopoulos, H., & Gammerman, A. (2009). Evolutionary conformal prediction for breast cancer diagnosis. In *2009 9th international conference on information technology and applications in biomedicine*, (pp. 1–4). https://doi.org/10.1109/ITAB.2009.5394447.

Laxhammar, Rikard., & Falkman, Göran. (2010). Conformal prediction for distribution-independent anomaly detection in streaming vessel data. In *Proceedings of the first international workshop on novel data stream pattern mining techniques*, StreamKDD '10, (pp. 47–55), New York, NY. ACM. ISBN 978-1-4503-0226-5. https://doi.org/10.1145/1833280.1833287.

Linusson, H., Norinder, U., Boström, H., Johansson, U., & Löfström, T. (2017). On the calibration of aggregated conformal predictors. In A. Gammerman, V. Vovk, Z. Luo, H. Papadopoulos (Eds.), *Proceedings of the sixth workshop on conformal and probabilistic prediction and applications*. *Proceedings of machine* *learning research* (Vol. 60, pp. 154–173). Stockholm, 13–16. PMLR. http://proceedings.mlr.press/v60/linusson17a.html.

Littell, R. C., & Folks, J. L. (1973). Asymptotic optimality of Fisher's method of combining independent tests II. *Journal of the American Statistical Association*, *68*(341), 193–194. https://doi.org/10.1080/01621459.1973.10481362.

Loughin, T. M. (2004). A systematic comparison of methods for combining p-values from independent tests. *Computational Statistics & Data Analysis*, *47*(3), 467–485.

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference*, (pp. 51–56).

Pesarin, F. (2001). *Multivariate permutation tests: With applications in biostatistics*. New York: Wiley. ISBN 9780471496700.

Poole, W., Gibbs, D. L., Shmulevich, I., Bernard, B., & Knijnenburg, T. A. (2016). Combining dependent p-values with an empirical adaptation of Brown's method. *Bioinformatics*, *32*(17), i430–i436. https://doi.org/10.1093/bioinformatics/btw438.

Pratt, J. W. (1959). Remarks on zeros and ties in the Wilcoxon signed rank procedure. *Journal of the American Statistical Association*, *54*, 655–667.

Shabbir, A., Verdoolaege, G., Vega, J., & Murari, A. (2015). ELM regime classification by conformal prediction on an information manifold. *IEEE Transactions on Plasma Science*, *43*(12), 4190–4199. https://doi.org/10.1109/TPS.2015.2489689. ISSN 0093-3813.

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421. ISSN 1532-4435.

Stouffer, E. A., Suchman, S. A., DeVinney, L. C., Star, S. A., & Williams, R. M, Jr. (1949). *The American soldier: Adjustment during army life*. Princeton: Princeton University Press.

Toccaceli, P., Nouretdinov, I., Gammerman, A. (2016). Conformal predictors for compound activity prediction. In A. Gammerman, Z. Luo, J. Vega, & V. Vovk, (Eds.), *In: Proceedings of 5th international symposium on conformal and probabilistic prediction with applications, COPA 2016, Madrid, Spain, April 20–22, 2016* (pp. 51–66). Cham: Springer International Publishing. ISBN 978-3-319-33395-3. https://doi.org/10.1007/978-3-319-33395-3_4.

Toccaceli, P., Nouretdinov, I., & Gammerman, A. (2017). Conformal prediction of biological activity of chemical compounds. *Annals of Mathematics and Artificial Intelligence*, *81*(1), 105–123. https://doi.org/10.1007/s10472-017-9556-8. ISSN 1573-7470.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., & Mueller, A. (2015). Scikit-learn: Machine learning without learning the machinery. *GetMobile*, *19*(1), 29–33. https://doi.org/10.1145/2786984.2786995.

Vovk, V. (2015). Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, *74*(1), 9–28. https://doi.org/10.1007/s10472-013-9368-4. ISSN 1573-7470.

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*. Secaucus, NJ: Springer New York Inc.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83.

Zaykin, D. V., Zhivotovsky, L. A., Czika, W., Shao, S., & Wolfinger, R. D. (2007). Combining p-values in large-scale genomics experiments. *Pharmaceutical Statistics*, *6*(3), 217–226. https://doi.org/10.1002/pst.304. ISSN 1539-1612.

Zaykin, D. V., Zhivotovsky, L. A., Westfall, P. H., & Weir, B. S. (2002). Truncated product method for combining p-values. *Genetic Epidemiology*, *22*(2), 170–185. https://doi.org/10.1002/gepi.0042. ISSN 1098-2272.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.