From Conformal to Probabilistic Prediction

Vladimir Vovk, Ivan Petej, and Valentina Fedorova

Computer Learning Research Centre,
Department of Computer Science,
Royal Holloway, University of London,
Egham, Surrey, UK
{volodya.vovk,ivan.petej,alushaf}@gmail.com

Abstract. This paper proposes a new method of probabilistic prediction, which is based on conformal prediction. The method is applied to the standard USPS data set and gives encouraging results.

1 Introduction

In essence, conformal predictors output systems of p-values: to each potential label of a test object a conformal predictor assigns the corresponding p-value, and a low p-value is interpreted as the label being unlikely. It has been argued, especially by Bayesian statisticians, that p-values are more difficult to interpret than probabilities; besides, in decision problems probabilities can be easily combined with utilities to obtain decisions that are optimal from the point of view of Bayesian decision theory. In this paper we will apply the idea of transforming p-values into probabilities (used in a completely different context in, e.g., [10], Sect. 9, and [7]) to conformal prediction: the p-values produced by conformal predictors will be transformed into probabilities.

The approach of this paper is as follows. It was observed in [12] that some criteria of efficiency for conformal prediction (called "probabilistic criteria") encourage using the conditional probability $Q(y \mid x)$ as the conformity score for an observation (x, y), Q being the data-generating distribution. In this paper we extend this observation to label-conditional predictors (Sect. 2).

Next we imagine that we are given a conformal predictor Γ that is nearly optimal with respect to a probabilistic criterion (such a conformal predictor might be an outcome of a thorough empirical study of various conformal predictors using a probabilistic criterion of efficiency). Essentially, this means that in the limit of a very large training set the p-value that Γ outputs for an observation (x, y) is a monotonic transformation of the conditional probability $Q(y \mid x)$ (Theorem 1 in Sect. 3).

Finally, we transform the p-values back into conditional probabilities using the distribution of p-values in the test set (Sect. 5). Following [10] and [7], we will say that at this step we *calibrate* the p-values into probabilities,

In Sect. 6 we give an example of a realistic situation where use of the techniques developed in this paper improves on a standard approach. The performance of the probabilistic predictors considered in that section is measured using standard loss functions, logarithmic and Brier (Sect. 4).

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 221–230, 2014.

Comparisons with Related Work

It should be noted that in the process of transforming p-values into probabilities suggested in this paper we lose a valuable feature of conformal prediction, its automatic validity. Our hope, however, is that the advantages of conformal prediction will translate into accurate probabilistic predictions.

There is another method of probabilistic prediction that is related to conformal prediction, Venn prediction (see, e.g., [13], Chap. 6, or [14]). This method does have a guaranteed property of validity (perhaps the simplest being Theorem 1 in [14]); however, the price to pay is that it outputs multiprobabilistic predictions rather than sharp probabilistic predictions. There are natural ways of transforming multiprobabilistic predictions into sharp probabilistic predictions (see, e.g., [14], Sect. 4), but such transformations, again, lead to the loss of the formal property of validity.

As preparation, we study label-conditional conformal prediction. For a general discussion of conditionality in conformal prediction, see [11]. Object-conditional conformal prediction has been studied in [5] (in the case of regression).

2 Criteria of Efficiency for Label-Conditional Conformal Predictors and Transducers

Let $\mathbf{X}$ be a measurable space (the *object space*) and $\mathbf{Y}$ be a finite set equipped with the discrete σ-algebra (the *label space*); the *observation space* is defined to be $\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$. A *conformity measure* is a measurable function A that assigns to every sequence $(z_1, \dots, z_n) \in \mathbf{Z}^*$ of observations a same-length sequence $(\alpha_1, \dots, \alpha_n)$ of real numbers and that is equivariant with respect to permutations: for any n and any permutation π of $\{1, \dots, n\}$,

$$(\alpha_1, \dots, \alpha_n) = A(z_1, \dots, z_n) \Longrightarrow \left(\alpha_{\pi(1)}, \dots, \alpha_{\pi(n)}\right) = A\left(z_{\pi(1)}, \dots, z_{\pi(n)}\right).$$

The *label-conditional conformal predictor* determined by A is defined by

$$\Gamma^\epsilon(z_1, \dots, z_l, x) := \{y \mid p^y > \epsilon\}, \tag{1}$$

where $(z_1, \dots, z_l) \in \mathbf{Z}^*$ is a training sequence, x is a test object, $\epsilon \in (0, 1)$ is a given *significance level*, and for each $y \in \mathbf{Y}$ the corresponding *label-conditional p-value* p^y is defined by

$$p^y := \frac{\left|\left\{i = 1, \dots, l+1 \mid y_i = y \ \ \& \ \ \alpha_i^y < \alpha_{l+1}^y\right\}\right|}{|\{i = 1, \dots, l+1 \mid y_i = y\}|} + \tau \frac{\left|\left\{i = 1, \dots, l+1 \mid y_i = y \ \ \& \ \ \alpha_i^y = \alpha_{l+1}^y\right\}\right|}{|\{i = 1, \dots, l+1 \mid y_i = y\}|}, \tag{2}$$

where τ is a random number distributed uniformly on the interval $[0, 1]$ and the corresponding sequence of *conformity scores* is defined by

$$(\alpha_1^y, \dots, \alpha_l^y, \alpha_{l+1}^y) := A(z_1, \dots, z_l, (x, y)).$$

It is clear that the system of *prediction sets* (1) output by a conformal predictor is nested, namely decreasing in ϵ.

The *label-conditional conformal transducer* determined by A outputs the system of p-values $(p^y \mid y \in \mathbf{Y})$ defined by (2) for each training sequence $(z_1, \dots, z_l)$ of observations and each test object x.

Four Criteria of Efficiency

Suppose that, besides the training sequence, we are also given a test sequence, and would like to measure on it the performance of a label-conditional conformal predictor or transducer. As usual, let us define the performance on the test set to be the average performance (or, equivalently, the sum of performances) on the individual test observations. Following [12], we will discuss the following four criteria of efficiency for individual test observations; all the criteria will work in the same direction: the smaller the better.

- The sum $\sum_{y \in \mathbf{Y}} p^y$ of the p-values; referred to as the *S criterion*. This is applicable to conformal transducers (i.e., the criterion is ϵ-independent).
- The size $|\Gamma^\epsilon|$ of the prediction set at a significance level ϵ; this is the *N criterion*. It is applicable to conformal predictors (ϵ-dependent).
- The sum of the p-values apart from that for the true label: the *OF* ("observed fuzziness") *criterion*.
- The number of false labels included in the prediction set Γ^ϵ at a significance level ϵ; this is the *OE* ("observed excess") *criterion*.

The last two criteria are simple modifications of the first two (leading to smoother and more expressive pictures). Equivalently, the S criterion can be defined as the arithmetic mean $\frac{1}{|\mathbf{Y}|} \sum_{y \in \mathbf{Y}} p^y$ of the p-values; the proof of Theorem 1 below will show that, in fact, we can replace arithmetic mean by any mean ([3], Sect. 3.1), including geometric, harmonic, etc.

3 Optimal Idealized Conformity Measures for a Known Probability Distribution

In this section we consider the idealized case where the probability distribution Q generating independent observations $z_1, z_2, \dots$ is known (as in [12]). The main result of this section, Theorem 1, is the label-conditional counterpart of Theorem 1 in [12]; the proof of our Theorem 1 is also modelled on the proof of Theorem 1 in [12]. In this section we assume, for simplicity, that the set $\mathbf{Z}$ is finite and that $Q(\{z\}) > 0$ for all $z \in \mathbf{Z}$.

An *idealized conformity measure* is a function $A(z, Q)$ of $z \in \mathbf{Z}$ and $Q \in \mathcal{P}(\mathbf{Z})$ (where $\mathcal{P}(\mathbf{Z})$ is the set of all probability measures on $\mathbf{Z}$). We will sometimes write the corresponding conformity scores as $A(z)$, as Q will be clear from the context. The *idealized smoothed label-conditional conformal predictor* corresponding to A outputs the following prediction set $\Gamma^\epsilon(x)$ for each object $x \in \mathbf{X}$ and each

significance level $\epsilon \in (0,1)$. For each potential label $y \in \mathbf{Y}$ for x define the corresponding *label-conditional p-value* as

$$p^y = p(x,y) := \frac{Q(\{(x',y) \mid x' \in \mathbf{X} \ \& \ A((x',y),Q) < A((x,y),Q)\})}{Q_{\mathbf{Y}}(\{y\})} + \tau \frac{Q(\{(x',y) \mid x' \in \mathbf{X} \ \& \ A((x',y),Q) = A((x,y),Q)\})}{Q_{\mathbf{Y}}(\{y\})} \tag{3}$$

(this is the idealized analogue of (2)), where $Q_{\mathbf{Y}}$ is the marginal distribution of Q on $\mathbf{Y}$ and τ is a random number distributed uniformly on $[0,1]$. The prediction set is

$$\Gamma^\epsilon(x) := \{y \in \mathbf{Y} \mid p(x,y) > \epsilon\}. \tag{4}$$

The *idealized smoothed label-conditional conformal transducer* corresponding to A outputs for each object $x \in \mathbf{X}$ the system of p-values $(p^y \mid y \in \mathbf{Y})$ defined by (3); in the idealized case we will usually use the alternative notation $p(x,y)$ for p^y.

Four Idealized Criteria of Efficiency

In this subsection we will apply the four criteria of efficiency that we discussed in the previous section to the idealized case of infinite training and test sequences; since the sequences are infinite, they carry all information about the data-generating distribution Q. We will write $\Gamma^\epsilon_A(x)$ for the $\Gamma^\epsilon(x)$ in (4) and $p_A(x,y)$ for the $p(x,y)$ in (3) to indicate the dependence on the choice of the conformity measure A. Let U be the uniform probability measure on the interval $[0,1]$.

An idealized conformity measure A is:

- *S-optimal* if $\mathbb{E}_{(x,\tau)\sim Q_{\mathbf{X}}\times U} \sum_y p_A(x,y) \le \mathbb{E}_{(x,\tau)\sim Q_{\mathbf{X}}\times U} \sum_y p_B(x,y)$ for any idealized conformity measure B, where $Q_{\mathbf{X}}$ is the marginal distribution of Q on $\mathbf{X}$;
- *N-optimal* if $\mathbb{E}_{(x,\tau)\sim Q_{\mathbf{X}}\times U} |\Gamma^\epsilon_A(x)| \le \mathbb{E}_{(x,\tau)\sim Q_{\mathbf{X}}\times U} |\Gamma^\epsilon_B(x)|$ for any idealized conformity measure B and any significance level ϵ;
- *OF-optimal* if

$$\mathbb{E}_{((x,y),\tau)\sim Q\times U} \sum_{y' \ne y} p_A(x,y') \le \mathbb{E}_{((x,y),\tau)\sim Q\times U} \sum_{y' \ne y} p_A(x,y')$$

  for any idealized conformity measure B;
- *OE-optimal* if

$$\mathbb{E}_{((x,y),\tau)\sim Q\times U} |\Gamma^\epsilon_A(x) \setminus \{y\}| \le \mathbb{E}_{((x,y),\tau)\sim Q\times U} |\Gamma^\epsilon_B(x) \setminus \{y\}|$$

  for any idealized conformity measure B and any significance level ϵ.

The *conditional probability (CP) idealized conformity measure* is

$$A((x,y),Q) := Q(y \mid x).$$

An idealized conformity measure A is a (label-conditional) *refinement* of an idealized conformity measure B if

$$B((x_1,y)) < B((x_2,y)) \Longrightarrow A((x_1,y)) < A((x_2,y)) \tag{5}$$

for all $x_1, x_2 \in \mathbf{Z}$ and all $y \in \mathbf{Y}$. (Notice that this definition, being label-conditional, is different from the one given in [12].) Let $\mathcal{R}(\mathrm{CP})$ be the set of all refinements of the CP idealized conformity measure. If C is a criterion of efficiency (one of the four discussed above), we let $\mathcal{O}(C)$ stand for the set of all C-optimal idealized conformity measures.

Theorem 1. $\mathcal{O}(\mathrm{S}) = \mathcal{O}(\mathrm{OF}) = \mathcal{O}(\mathrm{N}) = \mathcal{O}(\mathrm{OE}) = \mathcal{R}(\mathrm{CP})$.

Proof. We start from proving $\mathcal{R}(\mathrm{CP}) = \mathcal{O}(\mathrm{N})$. Fix a significance level ϵ. A smoothed confidence predictor at level ϵ is defined as a random set of observations $(x,y) \in \mathbf{Z}$; in other words, to each observation (x,y) is assigned the probability $P(x,y)$ that the observation will be outside the prediction set. Under the restriction that the sum of the probabilities $Q(x,y)$ of observations (x,y) outside the prediction set (defined as $\sum_x Q(x,y)P(x,y)$ in the smoothed case) is bounded by $\epsilon Q_{\mathbf{Y}}(y)$ for a fixed y, the N criterion requires us to make the sum of $Q_{\mathbf{X}}(x)$ for (x,y) outside the prediction set (defined as $\sum_x Q_{\mathbf{X}} P(x,y)$ in the smoothed case) as large as possible. It is clear that the set should consist of the observations with the smallest $Q(y \mid x)$ (by the usual Neyman–Pearson argument: cf. [4], Sect. 3.2).

Next we show that $\mathcal{O}(\mathrm{N}) \subseteq \mathcal{O}(\mathrm{S})$. Let an idealized conformity measure A be N-optimal. By definition,

$$\mathbb{E}_{x,\tau} \left| \Gamma^\epsilon_A(x) \right| \le \mathbb{E}_{x,\tau} \left| \Gamma^\epsilon_B(x) \right|$$

for any idealized conformity measure B and any significance level ϵ. Integrating over $\epsilon \in (0,1)$ and swapping the order of integrals and expectations,

$$\mathbb{E}_{x,\tau} \int_0^1 \left| \Gamma^\epsilon_A(x) \right| \, \mathrm{d}\epsilon \le \mathbb{E}_{x,\tau} \int_0^1 \left| \Gamma^\epsilon_B(x) \right| \, \mathrm{d}\epsilon. \tag{6}$$

Since

$$\left| \Gamma^\epsilon(x) \right| = \sum_{y \in \mathbf{Y}} 1_{\{p(x,y) > \epsilon\}},$$

we can rewrite (6), after swapping the order of summation and integration, as

$$\mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} \left(\int_0^1 1_{\{p_A(x,y) > \epsilon\}} \, \mathrm{d}\epsilon \right) \le \mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} \left(\int_0^1 1_{\{p_B(x,y) > \epsilon\}} \, \mathrm{d}\epsilon \right).$$

Since

$$\int_0^1 1_{\{p(x,y)>\epsilon\}} \, \mathrm{d}\epsilon = p(x,y),$$

we finally obtain

$$\mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} p_A(x,y) \le \mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} p_B(x,y).$$

Since this holds for any idealized conformity measure B, A is S-optimal.

The argument in the previous paragraph in fact shows that $\mathcal{O}(\mathrm{S}) = \mathcal{O}(\mathrm{N}) = \mathcal{R}(\mathrm{CP})$.

The equality $\mathcal{O}(\mathrm{S}) = \mathcal{O}(\mathrm{OF})$ follows from

$$\mathbb{E}_{x,\tau} \sum_{y} p(x,y) = \mathbb{E}_{(x,y),\tau} \sum_{y' \neq y} p(x,y') + \frac{1}{2},$$

where we have used the fact that $p(x,y)$ is distributed uniformly on $[0,1]$ when $((x,y),\tau) \sim Q \times U$ (see [13] and [12]).

Finally, we notice that $\mathcal{O}(\mathrm{N}) = \mathcal{O}(\mathrm{OE})$. Indeed, for any significance level ϵ,

$$\mathbb{E}_{x,\tau} \left| \Gamma^\epsilon(x) \right| = \mathbb{E}_{(x,y),\tau} \left| \Gamma^\epsilon(x) \setminus \{y\} \right| + (1-\epsilon),$$

again using the fact that $p(x,y)$ is distributed uniformly on $[0,1]$ and so $\mathbb{P}_{(x,y),\tau}(y \in \Gamma^\epsilon(x)) = 1 - \epsilon$. □

4 Criteria of Efficiency for Probabilistic Predictors

Given a training set $(z_1, \dots, z_l)$ and a test object x, a probabilistic predictor outputs a probability measure $P \in \mathcal{P}(\mathbf{Y})$, which is interpreted as its probabilistic prediction for the label y of x; we let $\mathcal{P}(\mathbf{Y})$ stand for the set of all probability measures on $\mathbf{Y}$. The two standard way of measuring the performance of P on the actual label y are the *logarithmic* (or *log*) *loss* $-\ln P(\{y\})$ and the *Brier loss*

$$\sum_{y' \in \mathbf{Y}} \left(1_{\{y'=y\}} - P(\{y'\}) \right)^2,$$

where 1_E stands for the indicator of an event E: $1_E = 0$ if E happens and $1_E = 0$ otherwise. The efficiency of probabilistic predictors will be measured by these two loss functions.

Suppose we have a test sequence $(z_{l+1}, \dots, z_{l+k})$, where $z_i = (x_i, y_i)$ for $i = l+1, \dots, l+k$, and we want to evaluate the performance of a probabilistic predictor (trained on a training sequence $z_1, \dots, z_l$) on it. In the next section we will use the *average log loss*

$$-\frac{1}{k} \sum_{i=l+1}^{l+k} \ln P_i(\{y_i\})$$

Algorithm 1. Conformal-type probabilistic predictor

Input: training sequence $(z_1, \dots, z_l) \in \mathbf{Z}^l$
Input: calibration sequence $(x_{l+1}, \dots, x_{l+k}) \in \mathbf{X}^k$
Input: test object x_0
Output: probabilistic prediction $P \in \mathcal{P}(\mathbf{Y})$ for the label of x_0
  for $y \in \mathbf{Y}$ **do**
    for each x_i in the calibration sequence find the p-value p_i^y by (2)
      (with $l+i$ in place of $l+1$)
    let g_y be the antitonic density on $[0,1]$ fitted to $p_{l+1}^y, \dots, p_{l+k}^y$
    find the p-value p_0^y by (2) (with 0 in place of $l+1$)
    for each $y \in \mathbf{Y}$, set $P'(\{y\}) := g_y(1)/g_y(p_0^y)$
  end for
  set $P(\{y\}) := P'(\{y\}) / \sum_{y'} P'(\{y'\})$ for each $y \in \mathbf{Y}$

and the *standardized Brier loss*

$$\sqrt{\frac{1}{k\left|\mathbf{Y}\right|} \sum_{i=l+1}^{l+k} \sum_{y' \in \mathbf{Y}} \left(1_{\{y'=y_i\}} - P_i(\{y'\})\right)^2},$$

where $P_i \in \mathcal{P}(\mathbf{Y})$ is the probabilistic prediction for x_i. Notice that in the binary case, $|\mathbf{Y}| = 2$, the average log loss coincides with the mean log error (used in, e.g., [14], (12)) and the standardized Brier loss coincides with the root mean square error (used in, e.g., [14], (13)).

5 Calibration of p-Values into Conditional Probabilities

The argument of this section will be somewhat heuristic, and we will not try to formalize it in this paper. Fix $y \in \mathbf{Y}$. Suppose that $q := P(y \mid x)$ has an absolutely continuous distribution with density f when $x \sim Q_{\mathbf{X}}$. (In other words, f is the density of the image of $Q_{\mathbf{X}}$ under the mapping $x \mapsto P(y \mid x)$.) For the CP idealized conformity measure, we can rewrite (3) as

$$p(q) := \int_0^q q' f(q') dq' \Big/ D \,, \tag{7}$$

where $D := Q_{\mathbf{Y}}(\{y\})$; alternatively, we can set $D := \int_0^1 q' f(q') dq'$ to the normalizing constant ensuring that $p(1) = 1$. To see how (7) is a special case of (3) for the CP idealized conformity measure, notice that the probability that $Y = y$ and $P(Y \mid X) \in (q', q' + dq')$, where $(X, Y) \sim f$, is $q' f(q') dq'$. In (7) we write $p(q)$ rather than p^y since p^y depends on y only via q.

We are more interested in the inverse function $q(p)$, which is defined by the condition

$$p = \int_0^{q(p)} q' f(q') dq' \Big/ D \,.$$

When $q \sim f$, we have

$$\mathbb{P}(p(q) \le a) = \mathbb{P}(q \le q(a)) = \int_0^{q(a)} f(q')dq'.$$

Therefore, when $q \sim f$, we have

$$\mathbb{P}(a \le p(q) \le a + da) = \int_{q(a)}^{q(a+da)} f(q')dq' \approx \frac{1}{q(a)} \int_{q(a)}^{q(a+da)} q' f(q')dq' = \frac{Dda}{q(a)},$$

and so

$$q(c) \approx D \Big/ \frac{\mathbb{P}(c \le p(q) \le c + dc)}{dc}.$$

This gives rise to the algorithm given as Algorithm 1, which uses real p-values (2) instead of the ideal p-values (3). The algorithm is transductive in that it uses a training sequence of labelled observations and a calibration sequence of unlabelled objects (in the next section we use the test sequence as the calibration sequence); the latter is used for calibrating p-values into conditional probabilities. Given all the p-values for the calibration sequence with postulated label y, find the corresponding antitonic density $g(p)$ (remember that the function $q(p)$ is known to be monotonic, namely isotonic) using Grenander's estimator (see [2] or, e.g., [1], Chap. 8). Use $D/g(p)$ as the calibration function, where $D := g(1)$ is chosen in such a way that a p-value of 1 is calibrated into a conditional probability of 1. (Alternatively, we could set D to the fraction of observations labelled as y in the training sequence; this approximates setting $D := Q_{\mathbf{Y}}(\{y\})$.) The probabilities produced by this procedure are not guaranteed to lead to a probability measure: the sum over y can be different from 1 (and this phenomenon has been observed in our experiments). Therefore, in the last line of Algorithm 1 we normalize the calibrated p-values to obtain genuine probabilities.

6 Experiments

In our experiments we use the standard USPS data set of hand-written digits. The size of the training set is 7291, and the size of the test set is 2007; however, instead of using the original split of the data into the two parts, we randomly split all available data (the union of the original training and test sets) into a training set of size 7291 and test set of size 2007. (Therefore, our results somewhat depend on the seed used by the random number generator, but the dependence is minor and does not affect our conclusions at all; we always report results for seed 0.)

A powerful algorithm for the USPS data set is the 1-Nearest Neighbour (1-NN) algorithm using tangent distance [8]. However, it is not obvious how this algorithm could be transformed into a probabilistic predictor. On the other hand, there is a very natural and standard way of extracting probabilities from support vector machines, which we will refer to it as *Platt's algorithm* in this paper: it is the combination of the method proposed by Platt [6] with pairwise coupling [15] (unlike our algorithm, which is applicable to multi-class problems directly, Platt's

Table 1. The performance of the two algorithms, Platt's (with the optimal values of parameters) and the conformal-type probabilistic predictor based on 1-Nearest Neighbour with tangent distance

algorithm	average log loss	standardized Brier loss
optimized Platt	0.06431	0.05089
conformal-type 1-NN	0.04958	0.04359

Table 2. The performance of Platt's algorithm with the polynomial kernels of various degrees for the cost parameter $C = 10$

degree	average log loss	standardized Brier loss
1	0.12681	0.07342
2	0.09967	0.06109
3	0.06855	0.05237
4	0.11041	0.06227
5	0.09794	0.06040

method is directly applicable only to binary problems). In this section we will apply our method to the 1-NN algorithm with tangent distance and compare the results to Platt's algorithm as implemented in the function `svm` from the `e1071` R package (for our multi-class problem this function calculates probabilities using the combination of Platt's binary method and pairwise coupling).

There is a standard way of turning a distance into a conformal predictor ([13], Sect. 3.1): namely, the conformity score α_i of the ith observation in a sequence of observations can be defined as

$$\frac{\min_{j:y_j \neq y_i} d(x_i, x_j)}{\min_{j \neq i: y_j = y_i} d(x_i, x_j)}, \tag{8}$$

where d is the distance; the intuition is that an object is considered conforming if it is close to an object labelled in the same way and far from any object labelled in a different way.

Table 1 compares the performance of the conformal-type probabilistic predictor based on the 1-NN conformity measure (8), where d is tangent distance, with the performance of Platt's algorithm with the optimal values of its parameters. The conformal predictor is parameter-free but Platt's algorithm depends on the choice of the kernel. We chose the polynomial kernel of degree 3 (since it is known to produce the best results: see [9], Sect. 12.2) and the cost parameter $C := 2.9$ in the case of the average log loss and $C := 3.4$ in the case of the standardized Brier loss (the optimal values in our experiments). (Reporting the performance of Platt's algorithm with optimal parameter values may look like data snooping, but it is fine in this context since we are helping our competitor.) Table 2 reports the performance of Platt's algorithm as function of the degree of the polynomial kernel with the cost parameter set at $C := 10$ (the dependence

on C is relatively mild, and $C = 10$ gives good performance for all degrees that we consider).

Acknowledgments. We thank the reviewer for useful comments. In our experiments we used the R package `e1071` (by David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, Chih-Chung Chang, and Chih-Chen Lin) and the implementation of tangent distance by Daniel Keysers. This work was partially supported by EPSRC (grant EP/K033344/1, first author) and Royal Holloway, University of London (third author).

References

1. Devroye, L.: A Course in Density Estimation. Birkhäuser, New York (1987)
2. Grenander, U.: On the theory of mortality measurement. Part II. Skandinavisk Aktuarietidskrift 39, 125–153 (1956)
3. Hardy, G.H., Littlewood, J.E., Pólya, G.: Inequalities, 2nd edn. Cambridge University Press, Cambridge (1952)
4. Lehmann, E.L.: Testing Statistical Hypotheses, 2nd edn. Springer, New York (1986)
5. Lei, J., Wasserman, L.: Distribution free prediction bands for nonparametric regression. Journal of the Royal Statistical Society B 76, 71–96 (2014)
6. Platt, J.C.: Probabilities for SV machines. In: Smola, A.J., Bartlett, P.L., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 61–74. MIT Press (2000)
7. Sellke, T., Bayarri, M.J., Berger, J.: Calibration of p-values for testing precise null hypotheses. American Statistician 55, 62–71 (2001)
8. Simard, P., LeCun, Y., Denker, J.: Efficient pattern recognition using a new transformation distance. In: Hanson, S., Cowan, J., Giles, C. (eds.) Advances in Neural Information Processing Systems, vol. 5, pp. 50–58. Morgan Kaufmann, San Mateo (1993)
9. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
10. Vovk, V.: A logic of probability, with application to the foundations of statistics (with discussion). Journal of the Royal Statistical Society B 55, 317–351 (1993)
11. Vovk, V.: Conditional validity of inductive conformal predictors. In: Hoi, S.C.H., Buntine, W. (eds.) JMLR: Workshop and Conference Proceedings, vol. 25, pp. 475–490 (2012); Asian Conference on Machine Learning. Full version: Technical report `arXiv:1209.2673` [cs.LG], `arXiv.org` e-Print archive (September 2012), The journal version: Machine Learning (ACML 2012 Special Issue) 92, 349–376 (2013).
12. Vovk, V., Fedorova, V., Gammerman, A., Nouretdinov, I.: Criteria of efficiency for conformal prediction, On-line Compression Modelling project (New Series), Working Paper 11 (April 2014) `http://alrw.net`
13. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
14. Vovk, V., Petej, I.: Venn–Abers predictors. In: Zhang, N.L., Tian, J. (eds.) Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, pp. 829–838. AUAI Press, Corvallis (2014), `http://auai.org/uai2014/proceedings/uai-2014-proceedings.pdf`
15. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research 5, 975–1005 (2004)

Aggregated Conformal Prediction

Lars Carlsson[1], Martin Eklund[2], and Ulf Norinder[3]

[1] AstraZeneca Research and Development, SE-431 83 Mölndal, Sweden
lars.a.carlsson@astrazeneca.com
[2] Department of Surgery, University of California San Francisco (UCSF), 1600 Divisadero St, San Francisco CA 94143, USA
martin.eklund@farmbio.uu.se
[3] H. Lundbeck A/S, Ottiliavej 9, 2500 Valby, Denmark
ulfn@lundbeck.com

Abstract. We present the aggregated conformal predictor (ACP), an extension to the traditional inductive conformal prediction (ICP) where several inductive conformal predictors are applied on the same training set and their individual predictions are aggregated to form a single prediction on an example. The results from applying ACP on two pharmaceutical data sets (CDK5 and GNRHR) indicate that the ACP has advantages over traditional ICP. ACP reduces the variance of the prediction region estimates and improves efficiency. Still, it is more conservative in terms of validity than ICP, indicating that there is room for further improvement of efficiency without compromising validity.

1 Introduction

Quantitative Structure-Activity Relationship (QSAR) modeling for predicting properties, e.g. solubility or toxicity, of chemical compounds using statistical learning techniques is a widespread approach within the pharmaceutical industry to prioritize compounds for experimental testing or to alert for potential toxicity. In making informed decisions based on predictions from QSAR models, the confidence in such predictions is of vital importance and conformal predictors have been successfully applied to the drug discovery setting [1]. In particular, we have shown that a Mondrian inductive conformal predictor is efficient (i.e. informative) and almost valid when applied to binary categorical data; exact validity was not achieved due to deviations from the exchangeability assumption [2]. Furthermore, we have demonstrated that an inductive conformal predictor (ICP) applied to regression data from the pharmaceutical industry was valid. However, for regression the prediction regions were in many cases as wide or wider than the possible ranges of the true responses (i.e. the range of the experimental assay that generates the true label). An important problem is thus to improve the efficiency of the ICPs when applied in the QSAR domain.

There are many ways to improve efficiency of conformal predictions, for example by using improved nonconformity scores, choosing different machine-learning methods, or using a transductive approach [3]. The transductive approach is the

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 231–240, 2014.

most appealing in terms of validity, but is often computationally costly and the nonconformity scores can be difficult to compute.

Another interesting approach to improve efficiency is the *cross conformal predictor* (CCP) [4]. Here the training data is divided into separate non-overlapping folds and each fold is used as a calibration set and the remainder of the data is used as a proper training set. This division allows for more data to be used for calibration and p-values are averaged over all folds. Similarly, the *bootstrap conformal predictor* (BCP) bootstraps datasets and uses the out-of-bag examples as a calibration set. p-values are then averages across all bootstrap replications.

In this paper we attempt to generalize the BCP and the CCP in what we term the *aggregated inductive conformal predictor* (ACP). We will empirically assess the ACP using data from the pharmaceutical domain and we show through a theoretical argument and experiments that ACP seems to have advantages over the standard ICP.

2 Aggregated Conformal Predictor

Consider the standard prerequisite for a description of a conformal predictor (CP), a bag of examples $\{z_1, \ldots, z_i, \ldots, z_l\}$ drawn from an exchangeable distribution Q. Each example $z_i = (x_i, y_i)$ can be described by its object $x_i \in \mathbf{X}$ and its label $y_i \in \mathbf{Y}$. The labels can be either categorical or continuous. For an inductive conformal predictor (ICP) the bag $\{z_i\}$ is partitioned into two different bags, one holding the proper training examples $\{z_1, \ldots, z_m\}$ and the other holding the calibration examples $\{z_{m+1}, \ldots, z_l\}$. The ICP p-value is then computed as

$$p = \frac{\mid \{j = m+1, ..., l : \alpha_j \geq \alpha_{l+1}\} \mid}{l - m + 1}$$

The prediction region of an ICP is determined by the "borderline" p-value, p_t, i.e. the smallest value p can obtain and still satisfy $p > \epsilon$. We can thus view p_t as the ϵth sample quantile (estimated from above)

$$p_t = U^{-1}_{l-m}(\epsilon),$$

where U_{l-m} is the empirical cumulative probability distribution of the p-values defined by

$$U_{l-m}(p) = \frac{1}{l - m + 1} \sum_{j=m+1}^{l} I(p_j < p),$$

where $I(\cdot) = 1$ if $p_j < p$ and $I(\cdot) = 0$ otherwise. We now introduce definitions of *Exchangeable resampling* and *Consistent resampling*, after which we define what we mean by an aggregated conformal predictor (ACP).

Definition 1 (Exchangeable resampling). *Let $\{z^*_1, \ldots, z^*_n\}$ be a bag of examples resampled from the empirical distribution Q_l. We call this resampling* exchangeable *if*

$$P\{(z_1^*, \ldots, z_n^*)\} = P\{(z_{\pi(1)}^*, \ldots, z_{\pi(n)}^*)\},$$

where π *is any permutation of* $\{1, ..., n\}$.

Definition 2 (Consistent resampling). *Let* $T = T(z_1, ..., z_l, Q)$ *be a statistic and* $T^* = T(z_1^*, ..., z_n^*, Q_l)$ *be an exchangeably resampled version of* T. *Further, let* G_l *and* G_l^* *be the probability distributions of* T *and* T^*, *respectively. We call the sampling process* consistent *(with respect to* T*) if*

$$\sup_z \mid G_l - G_l^* \mid \to 0 \text{ as } l \to \infty \text{ and } n \to \infty.$$

Definition 3 (ACP: Aggregated Conformal Predictor). *The following procedure is repeated* B *times, for* $b = 1, ..., B$*: Resample a bag* $\{z_1^*, \ldots, z_{n_b}^*\}$ *of examples from* $\{z_1, \ldots, z_l\}$ *using a consistent resampling procedure with respect to* α_t. *Compute the ICP p-value using the resampled bag,*

$$p_b^* = \frac{\mid \{j = m_b + 1, ..., n_b : \alpha_j^* \geq \alpha_{n_b+1}^*\} \mid}{n_b - m_b + 1}, \tag{1}$$

where α_j^* *are the nonconformity scores computed using* $\{z_1^*, \ldots, z_{n_b}^*\}$ *(*m_b *and* n_b *are indexed with* b *to make explicit that they may differ for different values of* b*). We define the ACP p-value as*

$$p_B = \frac{1}{B} \sum_{b=1}^{B} p_b^* \tag{2}$$

and the corresponding prediction region as

$$\Gamma^\epsilon(z_1, \ldots, z_l, x_{l+1}) := \{y | p_B > \epsilon\}. \tag{3}$$

A smoothed ACP can be defined analogously.

We note that the cross-conformal predictor and the bootstrapped conformal predictor suggested by Vovk [5] are two examples of ACPs.

Proposition 1. *The aggregated conformal predictor is conservatively valid.*

Proof. Since we use an exchangeable resampling procedure to construct the ACP and since an ICP is conservatively valid (Proposition 4.1, [3]), each resampled ICP in the ACP is conservatively valid (by symmetry). From this follows that the ACP also is conservatively valid.

Remark 1. Proposition 1 only holds unconditionally. The situation is different conditional on the particular dataset we have observed.

2.1 How Does the ACP Improve on the ICP?

For a p in an ICP, p_t is a hard threshold in the sense that the label y corresponding to p is either inside or outside the prediction region. Heuristically, the ACP averages over thresholds varying around p_t (since p_t^* based on a resampled bag $\{z_1^*, ..., z_l^*\}$ fluctuates around p_t), resulting in a smoothed threshold estimate with decreased variance compared to the estimate p_t.

We can use the method used by Bühlmann and Yu [6] to analyze this in a bit more detail. Consider the function

$$\delta(p) = I(p_t < p),$$

which indicates whether a p is smaller or larger than the borderline value p_t in an ICP (and thus if its corresponding label y either is inside or outside the prediction region). The sample quantile estimate p_t follows a normal distribution with mean $\epsilon = U^{-1}(\epsilon)$ and variance

$$\sigma^2 = \frac{(1-\epsilon)\epsilon}{(l-m+1)\left[f(q)\right]^2} = \frac{1-\epsilon}{(l-m+1)\epsilon},$$

where F is the population cumulative distribution function with density function f [7]. For a p in the neighborhood of ϵ

$$p = p(c) = \epsilon + c\sigma\sqrt{l-m+1} \tag{4}$$

we have the approximation

$$\delta(p(c)) \approx I(W < c), \quad W \sim N(0,1), \tag{5}$$

where W is the limiting random quantity from the asymptotic distribution of p_t (because of the construction of $p(c)$ in Equation (4)). For a fixed c, this is a hard threshold function of W. It follows that

$$\begin{aligned} &\mathrm{E}\left[\delta(p(c))\right] \to P(W < c) = \Phi(c) \text{ as } l \to \infty \text{ and } l - m \to \infty \\ &\mathrm{Var}\left[\delta(p(c))\right] \to \Phi(c)(1-\Phi(c)) \text{ as } l \to \infty \text{ and } l - m \to \infty, \end{aligned} \tag{6}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Note that the variance does not converge to zero; $\delta(p(c))$ assumes the values 0 and 1 with a positive probability even as l tends to infinity. However, for the ACP the situation looks different,

$$\begin{aligned} \delta_B(p(c)) &= \mathrm{E}^*\left[I(p_t^* \le p(c))\right] \\ &= \mathrm{E}^*\left[I\left(\sqrt{l-m+1}(p_t^* - p_t)/\sigma \le \sqrt{l-m+1}(p(c) - p_t)/\sigma\right)\right] \\ &= \Phi(\sqrt{l-m+1}(p(c) - p_t)) + o_p(1) \approx \Phi(c - W), \quad W \sim N(0,1). \end{aligned}$$

where the first approximation over the second equal sign follows because we by definition of the ACP have a consistent resampling process. Comparing with

Equation (5) for an ICP, the ACP produces a smoothed decision function of Z and therefore reduces variance. Again, following Bühlmann and Yu [6], we can study the case $p = p(0) = \epsilon$, i.e. when we are right at the population threshold and therefore has maximum variance. Then

$$\delta_B(p(0)) \to \Phi(-W) \sim U[0,1]$$

and, therefore,

$$\begin{aligned} &\mathrm{E}\left[\delta_B(p(0))\right] \to \mathrm{E}\left[U\right] = 1/2 \text{ as } l \to \infty \\ &\mathrm{Var}\left[\delta_B(p(0))\right] \to \mathrm{Var}(U) = 1/12 \text{ as } l \to \infty. \end{aligned} \tag{7}$$

Comparing Equation (6) to Equation (7), we see that the variance is reduced to one third for ACP compared to ICP.

3 Empirical Results of ACP

We used two different machine learning methods; the support vector machine (SVM) [8] implemented in the Java library version of libsvm [9] with a Gaussian radial basis kernel function

$$K(x, x') = \exp\left(-\gamma \|x - x'\|^2\right),$$

with $\gamma = 0.002$ and $C = 50$, and Random Forest (RF) [10], for which the default settings were used and each ensemble contained 100 trees.

Following Equation (16) in [11], we defined the nonconformity measure used in combination with SVM according to

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\exp\left(\hat{\mu}_i\right)}, \tag{8}$$

where $\hat{\mu}_i$ is the prediction of the value $\ln\left(|y_i - \hat{y}_i|\right)$ produced by a support vector regression machine trained on the proper training sets. After training the underlying SVM of the ICP, we calculate the residuals $|y_j - \hat{y}_j|$ for all proper training examples $j = 1, ..., m$ and train an SVM on the pairs $(x_i, \ln(|y_i - \hat{y}_i|))$. Measure (8) normalizes the absolute prediction error with the predicted accuracy of the SVM on a given example. The nonconformity measure used with RF was

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\hat{\nu}_i},$$

where $\hat{\nu}_i$ is the RF prediction of the value $|y_i - \hat{y}_i|$ with the same settings as for the other RF model.

Two public dataset (CDK5 and GNRHR) from the pharmaceutical domain was used [12]. The datasets consist of 230 and 198 examples, respectively. Each example (chemical compound) was described (characterised) by so-called signature descriptors [13] in the same way as described in [2].

The data was randomly split into two parts: A training set (80% of the original data) and a working set (20%). This procedure was repeated 50 times as to generate 50 training and working set, respectively. Furthermore, each training set was then, subsequently, randomly split into a proper training set (70% of the training set) and a calibration set (30%), similar to the 2:1 recommendation in [5]. This random selection of proper training and calibration examples was, in turn, repeated 100 times enabling the construction of 100 inductive conformal predictors for each working set. This sampling procedure is often called the $m-n$ sampling or non-replacement subsampling in the bootstrap literature and consistent with Definition 2 [14].

The results are presented in Tables 1- 8 and in Figures 1 and 2.

Table 1. The fraction of true labels within the prediction ranges for an ACP and a, for each run, randomly selected ICP at different significance levels for the 50 runs on the CDK data using SVM. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.4	0.6522	0.7391	0.7717	0.7717	0.8043	0.8913
ICP	0.4	0.5100	0.5622	0.5997	0.5990	0.6303	0.6826
ACP	0.3	0.7174	0.8261	0.8478	0.8474	0.8696	0.9348
ICP	0.3	0.6191	0.6535	0.6879	0.6896	0.7202	0.7876
ACP	0.2	0.8043	0.8913	0.9130	0.9170	0.9565	0.9783
ICP	0.2	0.7296	0.7712	0.7923	0.7989	0.8282	0.8848
ACP	0.1	0.8913	0.9565	0.9565	0.9643	0.9783	1.0000
ICP	0.1	0.8378	0.8626	0.8933	0.8890	0.9101	0.9537

Table 2. The fraction of true labels within the prediction ranges for an ACP and a, for each run, randomly selected ICP at different significance levels for the 50 runs on the GNRHR data using SVM. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.4	0.3590	0.5962	0.6667	0.6505	0.7179	0.8205
ICP	0.4	0.4297	0.5560	0.5960	0.5947	0.6569	0.7321
ACP	0.3	0.5128	0.6731	0.7179	0.7227	0.7628	0.8718
ICP	0.3	0.5215	0.6428	0.6736	0.6750	0.7208	0.7926
ACP	0.2	0.7179	0.7949	0.8205	0.8286	0.8462	0.9744
ICP	0.2	0.6782	0.7466	0.7749	0.7791	0.8126	0.8797
ACP	0.1	0.8205	0.8974	0.9231	0.9258	0.9487	1.0000
ICP	0.1	0.8072	0.8633	0.8745	0.8835	0.9104	0.9605

Table 3. The fraction of true labels within the prediction ranges for an ACP and a, for each run, randomly selected ICP at one significance level for the 50 runs on the CDK data using RF. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	0.7174	0.8478	0.8696	0.8696	0.9130	0.9565
ICP	0.2	0.6926	0.7786	0.8071	0.7982	0.8249	0.8930

Table 4. The fraction of true labels within the prediction ranges for an ACP and a, for each run, randomly selected ICP at one significance level for the 50 runs on the GNRHR data using RF. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	0.7179	0.8205	0.8718	0.8564	0.8974	1.0000
ICP	0.2	0.6854	0.7706	0.7919	0.8013	0.8271	0.9162

Table 5. The efficiency for an ACP and an ICP at one significance level for the 50 runs on the CDK5 data using SVM. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	0.5874	1.2190	1.6880	3.2000	3.3220	67.6700
ICP	0.2	0.1550	0.9128	1.2620	3.5760	1.8970	398.3000

Table 6. The efficiency for an ACP and an ICP at one significance level for the 50 runs on the CDK5 data using RF. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	0.4752	1.0060	1.2640	1.3750	1.5870	3.9780
ICP	0.2	0.0653	0.8659	1.1830	1.3680	1.6380	8.5950

4 Discussion

The results presented in Tables 1- 8 and in Figures 1 and 2 indicate that the ACP methodology has advantages over traditional ICP. The former adds stability and robustness to the predictions, which is particularly clear in Figures 1- 2 where the variance in the prediction ranges is smaller than from an individual ICP. This is of considerable importance within the pharmaceutical domain where precision as well as robustness in predictions are key elements for successful application in ongoing discovery projects. Although the traditional ICP is valid on average,

Table 7. The efficiency for an ACP and an ICP at one significance level for the 50 runs on the GNRHR data using SVM. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	1.2770	1.625	1.733	1.907	1.903	25.830
ICP	0.2	0.5486	1.457	1.704	1.957	2.000	62.400

Table 8. The efficency for an ACP and an ICP at one significance level for the 50 runs on the GNRHR data using RF. Here, Q_1 and Q_3 are the lower and upper quartile, respectively.

Method	ϵ	min	Q_1	median	mean	Q_3	max
ACP	0.2	0.7481	1.6050	1.9620	2.1290	2.5490	6.3040
ICP	0.2	0.07138	1.3940	1.8230	2.0980	2.4840	10.3100

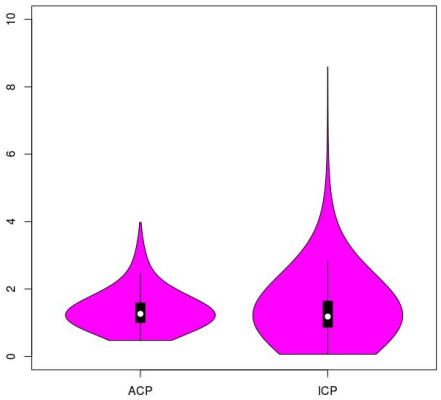


Fig. 1. The distribution of prediction interval size for an ACP and a randomly sampled ICP for all 50 runs on the CDK5 data. The results are shown at $\epsilon = 0.2$.

the variance is very large. This means that there is a relatively large proportion of very tight predictions regions that in fact are far from valid (for example, for the CDK5 data, an ICP with a confidence of 80% produces 46% error in the quartile with tightest prediction regions; the corresponding figure for the ACP is 21%. These tight prediction regions are offset by some prediction regions that are very wide (as wide or wider than the possible range of the response value) that are *always* valid, which produces a conformal predictor that is valid on average. Neither the too tight prediction regions that cannot be trusted nor non-informative regions are helpful for the researcher using the model.

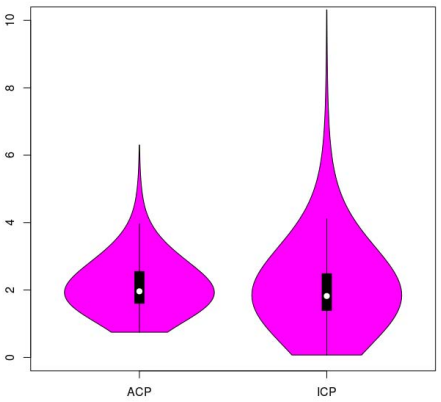


Fig. 2. The distribution of prediction interval size for an ACP and a randomly sampled ICP for all 50 runs on the GNRHR data. The results are shown at $\epsilon = 0.2$.

The results in Tables 1- 8 show that the ACP (as used in this paper) is too conservative on datasets with a relatively small number of examples, which is clear from Equations (1) and (2). This indicates that improved efficiency can be achieved without compromising the validity of the ACP, e.g. by more clever resampling (a large body of literature exists that address this problem, see e.g. [15]) or smoothing of p-values on either side of ϵ.

To conclude: We have introduced the ACP, a generalization of the BCP and CCP introduced by Vovk [5] and we have shown that it improves on classical ICP by reducing the variance in the estimated prediction regions (analogously to how a bagged predictor improves the prediction by reducing variance). ACP seems to represent a pragmatic and useful way forward for obtaining models and predictions with good precision and small prediction ranges.

Future ways to develop the ACP include to (i) study ACP for categorical labels (e.g. aggregation through voting); (ii) other resampling schemas.

References

1. Eklund, M., Norinder, U., Boyer, S., Carlsson, L.: Application of Conformal Prediction in QSAR. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 166–175. Springer, Heidelberg (2012)
2. Eklund, M., Norinder, U., Boyer, S., Carlsson, L.: The application of conformal prediction to the drug discovery process. Annals of Mathematics and Artificial Intelligence (2013)
3. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World, 1st edn. Springer (2005)

4. Vovk, V.: Cross-conformal predictors. Annals of Mathematics and Artificial Intelligence, 1–20 (2013)
5. Vovk, V.: Cross-conformal predictor. Working Paper 6 (2013), http://alrw.net
6. Bühlmann, P., Yu, B.: Analyzing bagging. The Annals of Statistics 30(4), 927–961 (2002)
7. Ruppert, D.: Statistics and Data Analysis for Financial Engineering, 1st edn. Springer Texts in Statistics. Springer, Berlin (2010)
8. Vapnik, V.N.: Statistical learning theory, 1 edn. Wiley (1998)
9. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
11. Papadopoulos, H., Haralambous, H.: Reliable prediction intervals with regression neural networks. Neural Networks 24(8), 842–851 (2011)
12. Chen, H., Carlsson, L., Eriksson, M., Varkonyi, P., Norinder, U., Nilsson, I.: Beyond the scope of free-wilson analysis: Building interpretable qsar models with machine learning algorithms. Journal of Chemical Information and Modeling 53, 1324–1336 (2013)
13. Faulon, J.-L., Collins, M.J., Carr, R.D.: The signature molecular descriptor. 4. Canonizing molecules using extended valence sequences. J. Chem. Inf. Comput. Sci. 44(2), 427–436 (2004)
14. Politis, D.N., Romano, J.P., Wolf, M.: Subsampling. Springer, New York (1999)
15. Egloff, D., Leippold, M.: Quantile estimation with adaptive importance sampling. The Annals of Statistics 38(2), 1244–1278 (2010)

A Cross-Conformal Predictor for Multi-label Classification

Harris Papadopoulos

Computer Science and Engineering Department, Frederick University,
7 Y. Frederickou St., Palouriotisa, Nicosia 1036, Cyprus
h.papadopoulos@frederick.ac.cy

Abstract. Unlike the typical classification setting where each instance is associated with a single class, in multi-label learning each instance is associated with multiple classes simultaneously. Therefore the learning task in this setting is to predict the subset of classes to which each instance belongs. This work examines the application of a recently developed framework called Conformal Prediction (CP) to the multi-label learning setting. CP complements the predictions of machine learning algorithms with reliable measures of confidence. As a result the proposed approach instead of just predicting the most likely subset of classes for a new unseen instance, also indicates the likelihood of each predicted subset being correct. This additional information is especially valuable in the multi-label setting where the overall uncertainty is extremely high.

1 Introduction

Most machine learning research on classification deals with problems in which each instance is associated with a single class y from a set of classes $\{Y_1, \ldots, Y_c\}$. As opposed to this standard setting, in *multi-label classification* each instance can belong to multiple classes, so that each instance is associated with a set of classes $\psi \subseteq \{Y_1, \ldots, Y_c\}$, called a *labelset*. There are many real-world problems in which such a setting is natural. For instance in the categorization of news articles an article discussing the positions of political parties on the educational system of a country can be classified as both politics and education. Although until recently the main multi-label classification application was the categorization of textual data, in the last few years an increasing number of new applications started to attract the attention of more researchers to this setting. Such applications include the semantic annotation of images and videos, the categorization of music into emotions, functional genomics, proteomics and directed marketing.

As a result of the increasing attention to the multi-label classification setting, many new machine learning techniques have been recently developed to deal with problems of this type, see e.g. [2, 8–11]. However, like most machine learning methods, these techniques do not produce any indication about the likelihood of each of their predicted labelsets being correct. Such an indication though can be very helpful in deciding how much to rely on each prediction, especially since

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 241–250, 2014.

the certainty of predictions may vary to a big degree between instances. To address this problem this paper examines the application of a recently developed framework for providing reliable confidence measures to predictions, called *Conformal Prediction* (CP) [7], to the multi-label classification setting. Specifically it follows the newly proposed *Cross-Conformal Prediction* (CCP) [6] version of the framework, which allows it to overcome the prohibitively large computational overhead of the original CP. The proposed approach computes a p-value for each of the possible labelsets, which can be used either for accompanying each prediction with confidence measures that indicate its likelihood of being correct, or for producing sets of labelsets that are guaranteed to contain the true labelset with a frequency equal to or higher than a required level of confidence.

In the remaining paper, a description of the general idea behind CP and of the CCP version of the framework, is first provided in Section 2. Section 3 defines the proposed approach while Section 4 presents the experiments performed and the obtained results. Finally Section 5 gives the conclusions of this work.

2 Conformal and Cross-Conformal Prediction

Typically in classification we are given a set of training examples $\{z_1, \ldots, z_l\}$, where each $z_i \in \mathcal{Z}$ is a pair (x_i, y_i) consisting of a vector of attributes $x_i \in \mathbb{R}^d$ and the classification $y_i \in \{Y_1, \ldots, Y_c\}$. We are also given a new unclassified example x_{l+1} and our task is to state something about our confidence in each possible classification of x_{l+1} without assuming anything more than that all (x_i, y_i), $i = 1, 2, \ldots$, are independent and identically distributed.

The idea behind CP is to assume every possible classification Y_j of the example x_{l+1} and check how likely it is that the extended set of examples

$$\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y_j)\} \tag{1}$$

is i.i.d. This in effect will correspond to the likelihood of Y_j being the true label of the example x_{l+1} since this is the only unknown value in (1).

First a function A called *nonconformity measure* is used to map each pair (x_i, y_i) in (1) to a numerical score

$$\alpha_i = A(\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y_j)\}, (x_i, y_i)), \quad i = 1, \ldots, l, \tag{2a}$$

$$\alpha_{l+1}^{Y_j} = A(\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y_j)\}, (x_{l+1}, Y_j)), \tag{2b}$$

called the *nonconformity score* of instance i. This score indicates how nonconforming, or strange, it is for i to belong in (1). In effect the nonconformity measure is based on a conventional machine learning algorithm, called the *underlying algorithm* of the corresponding CP and measures the degree of disagreement between the actual label y_i and the prediction $\hat{y}_i$ of the underlying algorithm, after being trained on (1). The nonconformity measure for multi-label learning used in this work is defined in Section 3.

The nonconformity score $\alpha_{l+1}^{Y_j}$ is then compared to the nonconformity scores of all other examples to find out how unusual (x_{l+1}, Y_j) is according to the

nonconformity measure used. This comparison is performed with the function

$$p((x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y_j)) = \frac{|\{i = 1, \ldots, l, l+1 : \alpha_i^{Y_j} \geq \alpha_{l+1}^{Y_j}\}|}{l+1}, \quad (3)$$

the output of which is called the p-value of Y_j, also denoted as $p(Y_j)$. An important property of (3) is that $\forall \delta \in [0, 1]$ and for all probability distributions P on $\mathcal{Z}$,

$$P^{l+1}\{((x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, y_{l+1})) : p(y_{l+1}) \leq \delta\} \leq \delta; \quad (4)$$

a proof can be found in [7]. According to this property, if $p(Y_j)$ is under some very low threshold, say 0.05, this means that Y_j is highly unlikely as the probability of such an event is at most 5% if (1) is i.i.d.

There are two standard ways to use the p-values of all possible classifications for producing the output of a CP:

- Predict the classification with the highest p-value and output one minus the second highest p-value as confidence to this prediction and the p-value of the predicted classification (i.e. the highest p-value) as credibility.
- Given a confidence level $1 - \delta$, output the prediction set $\{Y_j : p(Y_j) > \delta\}$.

In the first case, confidence is an indication of how likely the prediction is of being correct compared to all other possible classifications, whereas credibility indicates how suitable the training set is for the particular instance; specifically a very low credibility value indicates that the particular instance does not seem to belong to any of the possible classifications. In the second case, the prediction set will not contain the true label of the instance with at most δ probability.

The important drawback of the above process is that since the last example in (1) changes for every possible classification, the underlying algorithm needs to be trained c times. Moreover the whole process needs to be repeated for every test example. This makes it extremely computationally inefficient in many cases and especially in a multi-label setting where there are 2^n possible labelsets for n classes, or $2^n - 1$ if we exclude the empty labelset.

To overcome this computational inefficiency problem an inductive version of the framework was proposed in [3] and [4] called *Inductive Conformal Prediction* (ICP). ICP is based on the same theoretical foundations described above, but follows a modified version of the approach, which allows it to train the underlying algorithm only once. This is achieved by dividing the training set into two smaller sets, the *proper training set* and the *calibration set*. The proper training set is then used for training the underlying algorithm of the ICP and only the examples in the calibration set are used for calculating the p-value of each possible classification for every test example.

Although ICP is much more computationally efficient than the original CP approach, the fact that it does not use the whole training set for training the underlying algorithm and for calculating its p-values results in lower informational efficiency. That is the resulting prediction sets might be larger than the ones produced by the original CP approach. Cross-Conformal Prediction, which

was recently proposed in [6], tries to overcome this problem by combining ICP with cross-validation. Specifically, CCP partitions the training set in K subsets (folds) $S_1, \ldots, S_K$ and calculates the nonconformity scores of the examples in each subset S_k and of (x_{l+1}, Y_j) for each possible classification Y_j as

$$\alpha_i = A(\cup_{m \neq k} S_m, (x_i, y_i)), \quad i \in S_k, \quad m = 1, \ldots, K, \tag{5a}$$

$$\alpha_{l+1}^{Y_j,k} = A(\cup_{m \neq k} S_m, (x_{l+1}, Y_j)), \quad m = 1, \ldots, K, \tag{5b}$$

where A is the given nonconformity measure. Note that for (x_{l+1}, Y_j) K nonconformity scores $\alpha_{l+1}^{Y_j,k}$, $k = 1, \ldots, K$ are calculated, one with each of the K folds. Now the p-value for each possible classification Y_j is computed as

$$p(Y_j) = \frac{\sum_{k=1}^{K} |\{(x_i, y_i) \in S_k : \alpha_i \geq \alpha_{l+1}^{Y_j,k}\}| + 1}{l+1}. \tag{6}$$

The CCP version of the framework was chosen to be followed here due to its big advantage in computational efficiency over CP, since it needs to train the underlying algorithm only K times, and its advantage over ICP since it utilizes the whole training set for producing its p-values. As opposed to CP and ICP, the validity of which has been proven theoretically, at the moment there are no theoretical results about the validity of CCP. However in [6] its outputs have been shown to be empirically valid. The same is shown in the experimental results of this work, presented in Section 4.

3 ML-RBF Cross-Conformal Predictor

This section describes the proposed approach, which in effect comes down to the definition of a suitable nonconformity measure for multi-label classification and its use with the CCP framework. In order to use the CCP framework in the multi-label setting, the set of possible classifications $\{Y_1, \ldots, Y_c\}$ is replaced by the powerset $\mathcal{P}(\{Y_1, \ldots, Y_n\})$, where n is the number of the original classes of the problem. In the experiments that follow RBF Neural Networks for multi-label learning (ML-RBF) [8] is used as underlying algorithm as it seems to be one of the best performing algorithms designed specifically for multi-label problems. However the proposed approach is general and can be used with any other method which gives scores for each class.

After being trained on a given training set, for every test example ML-RBF produces a score for each possible class of the task at hand. It then outputs as its prediction the labelset containing all classes with score higher than zero; higher score indicates higher chance of the class to be in the labelset. In order to use the scores produced by ML-RBF for computing the nonconformity scores of the proposed Cross-Conformal Predictor, the former were transformed to the range $[0, 1]$ with the logistic sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}}.$$

The nonconformity measure (5) for the multi-label CCP can now be defined based on the transformed outputs of ML-RBF for (x_i, ψ_i) after being trained on $\cup_{m \neq k} S_m$, $m = 1, \ldots, K$ as

$$\alpha_i = \sum_{j=1}^{n} |t_i^j - o_i^j|^d, \tag{7}$$

where n is the number of possible classes, t_i^j is 1 if $Y_j \in \psi_i$ and 0 otherwise, and o_i^j is the transformed output of the ML-RBF corresponding to class Y_j. Finally d is a parameter of the algorithm which controls the sensitivity of the nonconformity measure to small differences in o_i^j when $|t_i^j - o_i^j|$ is small as opposed to when it is large. This nonconformity measure takes into account the distance of all outputs from the true values. Note that in the case of test examples t_i^j is the corresponding value for each assumed labelset.

The nonconformity measure (7) takes into account only the outputs of the ML-RBF for each instance. However, it is very strange to have a pair of labels in a labelset that have never appeared together in the training set. So (7) was extended to take into account the occurrence of each pair of labels in the training set $\cup_{m \neq k} S_m$, $m = 1, \ldots, K$. This extended nonconformity measure is defined as:

$$\alpha_i = \sum_{j=1}^{n} |t_i^j - o_i^j|^d + \lambda \sum_{1 \leq j < r \leq n} t_i^j t_i^r \mu_{j,r}, \tag{8}$$

where $\mu_{j,r}$ is 0 if the labels Y_j and Y_r have been observed together in the labelset of at least one instance of the training set and 1 otherwise. In effect the additional part of this nonconformity measure adds λ to the nonconformity score of an example for each pair of labels in the labelset of the example ($t_i^j t_i^r = 1$) which have never been observed together in any instance of the training set ($\mu_{j,r} = 1$). The parameter λ adjusts the sensitivity to this part of the measure. A high value of λ makes this part of the measure dominate when a pair of labels that has never been observed before exists in the labelset of the example.

The complete proposed approach is derived by plugging in (8) as A in (5) and computing the p-value of each possible labelset with (6). The resulting p-values can be used in either of the two ways described in Section 2.

4 Experiments and Results

4.1 Data Sets

To evaluate the performance of the proposed approach two data sets from different application domains were used, one from the semantic scene analysis domain and one from the bioinformatics domain. The first data set, *scene* [1], is concerned with the semantic classification of pictures into one or more of the classes: beach, sunset, foliage, field, mountain and urban. It consists of 1211 training and 1196 test examples, each described by 294 features. The second data set, *yeast*

[2], is concerned with predicting the functional classes of genes in the Yeast Saccharomyces cerevisiae. Each gene is described by the concatenation of microarray expression data and a phylogenetic profile, and is associated with a set of 14 functional classes. The data set contains 1500 genes as training set and 917 genes as test set, each described by 103 features. Both data sets were obtained from the website of the Mulan library [5].

4.2 Single Prediction Evaluation

The first set of experiments evaluates the quality of the single predictions produced by the ML-RBF CCP and compares it to that of its underlying method and those of three other popular multi-label techniques, namely BP-MLL [10], ML-kNN [11] and ML-NB [9]. Four evaluation measures for multi-label classification were used. The first is Hamming Loss (HL), which is the most popular measure for multi-label problems, defined as:

$$HL = \frac{1}{g} \sum_{i=l+1}^{l+g} \frac{|\psi_i \triangle \hat{\psi}_i|}{n}, \tag{9}$$

where $\{(x_{l+1}, \psi_{l+1}), \ldots, (x_{l+g}, \psi_{l+g})\}$ are the test examples, $\hat{\psi}_i$ is the predicted labelset for example i and $\triangle$ is the symmetric difference between two sets. The second measure is Classification Accuracy (CA) defined as:

$$CA = \frac{1}{g} \sum_{i=l+1}^{l+g} I(\psi_i = \hat{\psi}_i), \tag{10}$$

where $I(true) = 1$ and $I(false) = 0$. This measure is rather strict as it requires the predicted and true labelsets to be identical. The third and fourth measures are the *macro averaged* and *micro averaged* F-measure, which is the harmonic mean of precision and recall. The F-measure for a single label is defined as:

$$F(tp, tn, fp, fn) = \frac{2tp}{2tp + fp + fn}, \tag{11}$$

where tp is the number of true positives, tn the number of true negatives, fp the number of false positives and fn the number of false negatives. In the multi-label case, if tp_j, tn_j, fp_j and fn_j are the same values for each label Y_j, then its macro averaged version is defined as:

$$F_{macro} = \frac{1}{n} \sum_{j=1}^{n} F(tp_j, tn_j, fp_j, fn_j). \tag{12}$$

The micro averaged version of the F-measure is defined as:

$$F_{micro} = F\left(\sum_{j=1}^{n} tp_j, \sum_{j=1}^{n} tn_j, \sum_{j=1}^{n} fp_j, \sum_{j=1}^{n} fn_j\right). \tag{13}$$

Table 1. Performance of the proposed approach and comparison to that of other multi-label algorithms on the scene data set

	Evaluation Measure			
Algorithm	HL	CA	F_{macro}	F_{micro}
ML-RBF CCP with $\lambda = 0$	0.0928	0.6798	**0.7417**	**0.7363**
ML-RBF CCP with $\lambda = 1$	**0.0927**	**0.6831**	0.7410	0.7358
Orgiginal ML-RBF	0.0959	0.5468	0.6922	0.6890
BP-MLL	0.2903	0.1630	0.0509	0.1665
ML-kNN	0.0953	0.6012	0.7189	0.7183
ML-NB	0.1309	0.4105	0.6230	0.6221

Table 2. Performance of the proposed approach and comparison to that of other multi-label algorithms on the yeast data set

	Evaluation Measure			
Algorithm	HL	CA	F_{macro}	F_{micro}
ML-RBF CCP with $\lambda = 0$	**0.1954**	0.1821	**0.3896**	**0.6432**
ML-RBF CCP with $\lambda = 1$	**0.1954**	0.1821	**0.3896**	**0.6432**
Orgiginal ML-RBF	0.1970	**0.1865**	0.3891	0.6407
BP-MLL	0.2272	0.0960	0.3047	0.6212
ML-kNN	0.1980	0.1658	0.3567	0.6360
ML-NB	0.2115	0.1254	0.3428	0.6152

Tables 1 and 2 present the performance of the proposed approach together with that of its underlying algorithm and that of BP-MLL, ML-kNN and ML-NB on the scene and yeast data sets respectively. The best values for each measure are highlighted in bold. For ML-RBF the fraction parameter was set to 0.01 and the scaling factor to 1 as in [8]. In the case of the CCP the number of folds for each data set was chosen so that each fold contained approximately 100 training instances, therefore 12 folds were used for the scene data set and 15 folds for the yeast dataset. The parameter d of the nonconformity measure (8) was set to 4, which seems to be a good choice based on the performed experiments, while for λ the two extreme values of 0 and 1 were used. Setting λ to 0 in effect corresponds to nonconformity measure (7) and setting it to 1 makes the nonconformity score of any labelset containing a pair of classes that has never been observed in the training set always higher than others. For BP-MLL the number of hidden neurons was set to 20% of the input dimensionality, the learning rate to 0.05 and the training epochs to 100 as in [10]. For ML-kNN the number of nearest neighbours was set to 10 and the smoothing parameter to 1 as in [11]. For ML-NB the percentage of remaining features after PCA was set to 0.3 as in [9].

The results in these tables show that not only the proposed approach provides important additional information about the likelihood of each of its predictions being correct, but it also outperforms its underlying algorithm and the three other popular multi-label techniques. The only exception is the classification

Table 3. Prediction set sizes and error rates at the 95%, 90% and 80% confidence levels for the scene data set

# of labelsets	With $\lambda = 0$ Confidence Level 95%	90%	80%	With $\lambda = 1$ Confidence Level 95%	90%	80%
1	0.00%	2.34%	54.93%	6.77%	18.23%	62.63%
2	0.08%	25.50%	34.20%	9.62%	32.69%	29.43%
3 to 2^2	25.50%	65.64%	10.87%	43.31%	45.07%	7.94%
(2^2+1) to 2^3	72.74%	6.52%	0.00%	40.05%	4.01%	0.00%
(2^3+1) to 2^4	1.67%	0.00%	0.00%	0.25%	0.00%	0.00%
Errors	3.76%	9.28%	21.07%	3.60%	9.28%	20.99%

accuracy of the ML-RBF in the case of the yeast data set, but even in this case the accuracy of the proposed approach with both values of λ is very close. Comparing the performance of the ML-RBF CCP with the two different λ values one can see that in the case of the scene data there is only a negligible difference while in the case of the yeast data the performance remains the same for all evaluation measures. Therefore the value of λ does not have any important effect on the performance of the single predictions produced by the ML-RBF CCP. It does however affect the quality of the resulting p-values as will be shown in the next subsection.

4.3 Prediction Region Evaluation

The main advantage of the proposed approach over other multi-label techniques is the production of a p-value for each possible labelset of a new unseen instance, which can be translated either to confidence and credibility measures for its prediction or to prediction sets that are guaranteed to contain the true labelset at a required confidence level. This subsection examines the informativeness and reliability of the resulting prediction sets and consequently of the computed p-values and confidence measures. More specifically given a required level of confidence $1 - \delta$, the ML-RBF CCP produces a set of labelsets that has at most δ chance of not containing the true labelset of the unseen instance. The informativeness of this set of labelsets can be assessed in terms of its size, while its reliability can be assessed by the percentage of cases for which it does not contain the true labelset, this percentage should be less than or very near δ.

Tables 3 and 4 present the results of the proposed approach in this setting for the scene and yeast data sets respectively with the nonconformity measure parameter λ set to 0 and 1. The same parameters reported in Subsection 4.2 were used. The two tables report the sizes of the prediction sets produced for the 95%, 90% and 80% confidence levels together with the observed error percentages, i.e. the percentages of prediction sets that did not contain the true labelset.

Table 3 reports the results for the scene data set in terms of the percentage of prediction sets containing only 1, 2, 3 to 4, 5 to 8 and 9 to 16 labelsets for each

Table 4. Prediction set sizes and error rates at the 95%, 90% and 80% confidence levels for the yeast data set

# of labelsets	With $\lambda = 0$ Confidence Level			With $\lambda = 1$ Confidence Level		
	95%	90%	80%	95%	90%	80%
$(2^6 + 1)$ to 2^7	0.00%	0.00%	1.42%	0.00%	0.00%	1.42%
$2^7 < l \leq 2^8$	0.00%	0.11%	3.71%	0.00%	0.22%	4.58%
$2^8 < l \leq 2^9$	0.55%	3.82%	14.07%	0.76%	4.36%	18.65%
$2^9 < l \leq 2^{10}$	3.05%	7.09%	60.96%	3.93%	10.14%	60.32%
$2^{10} < l \leq 2^{11}$	8.94%	34.46%	19.85%	13.85%	45.58%	15.05%
$2^{11} < l \leq 2^{12}$	38.71%	52.89%	0.00%	57.58%	39.48%	0.00%
$2^{12} < l \leq 2^{13}$	48.64%	1.64%	0.00%	23.88%	0.22%	0.00%
$2^{13} < l \leq 2^{14}$	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%
Errors	4.69%	9.38%	19.85%	4.80%	9.60%	19.96%

confidence level; there was no prediction set containing more than 16 labelsets out of the possible 63. The last row of the table reports the percentage of errors observed for each confidence level. Comparing the results obtained with the nonconformity measure parameter λ set to 0 with those obtained with $\lambda = 1$, one can see that the latter produces much more informative prediction sets. Taking into account the classification accuracy of this data set (68.31%) and the large number of possible labelsets, the resulting prediction sets are quite tight. One can be 95% confident in about 60% of the test instances by considering less than 4 out of the possible 63 labelsets. By reducing the required confidence to 90% one can be certain in a single labelset for about 18% of the test instances and between one or two labelsets for about half the test instances. Finally with a confidence level of 80% a single labelset is given for more than 60% of the test instances. In terms of empirical reliability, only the percentages of errors for the 80% confidence level are slightly higher than the required significance level, which can be attributed to statistical fluctuations.

Table 4 presents the same results for the yeast data set. In this case the prediction sets contained a much higher number of labelsets so the table reports the percentage of prediction sets containing between 2^i+1 and 2^{i+1} labelsets with $i = 6, \ldots, 13$ for each confidence level; there were no prediction sets containing less than 2^6 labelsets. The rather big size of the resulting prediction sets is not strange baring in mind the very low classification accuracy of this data set, which is only 18.65%. Comparing the results obtained with the nonconformity measure parameter λ set to 0 with those obtained with $\lambda = 1$, again shows the superiority of nonconformity measure (8), as it produces smaller prediction sets. Considering the high difficulty of the particular task one can say that the resulting prediction sets are quite informative. The number of labelsets needed to satisfy the 80% confidence level is 1/16th or less ($\leq 2^{10}$) of all the possible labelsets for 85% of the test instances. Finally in terms of empirical reliability, the percentage of errors observed is in all cases below the required significance level.

5 Conclusions

This work examined the application of the conformal prediction framework to the multi-label setting. Unlike the other techniques developed for multi-label problems, the proposed approach accompanies each of its predictions with reliable measures of confidence. Experimental results on two popular multi-label data sets have shown that not only the proposed approach provides important additional information for each prediction, but it also outperforms other popular multi-label techniques. Furthermore its confidence measures have been shown to be informative and reliable. The provision of confidence measures can be very helpful in practical applications, considering the high uncertainty that exists in this setting.

Future work includes the development of additional nonconformity measures and the experimentation with more multi-label data. In addition generating separate p-values for each class and combining them for obtaining the p-value of each labelset could also be examined as an alternative. Finally the possibility of generating a ranking of the possible classes for each instance would also be a good addition to multi-label CP.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37, 1757–1771 (2004)
2. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems 14, pp. 681–687. MIT Press (2002)
3. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive confidence machines for regression. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 345–356. Springer, Heidelberg (2002)
4. Papadopoulos, H., Vovk, V., Gammerman, A.: Qualified predictions for large data sets in the case of pattern recognition. In: Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA 2002), pp. 159–163. CSREA Press (2002)
5. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
6. Vovk, V.: Cross-conformal predictors. Annals of Mathematics and Artificial Intelligence (2013), http://dx.doi.org/10.1007/s10472-013-9368-4
7. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
8. Zhang, M.L.: ML-RBF: RBF neural networks for multi-label learning. Neural Processing Letters 29(2), 61–74 (2009)
9. Zhang, M.L., Peña, J.M., Robles, V.: Feature selection for multi-label naive bayes classification. Information Sciences 179(19), 3218–3229 (2009)
10. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)
11. Zhang, M.L., Zhou, Z.H.: ML-kNN: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)

SVM Venn Machine with k-Means Clustering

Chenzhe Zhou, Ilia Nouretdinov, Zhiyuan Luo, and Alex Gammerman

Computer Learning Research Centre,
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK

Abstract. In this paper, we introduce a new method of designing Venn Machine taxonomy based on Support Vector Machines and k-means clustering for both binary and multi-class problems. We compare this algorithm to some other multi-probabilistic predictors including SVM Venn Machine with homogeneous intervals and a recently developed algorithm called Venn-ABERS predictor. These algorithms were tested on a range of real-world data sets. Experimental results are presented and discussed.

Keywords: Venn Machine, Support Vector Machine, k-means clustering.

1 Introduction

Classification is one of the major tasks in machine learning. It gives predictions for the new objects based on known properties learned from the training data set. However, most algorithms could only give single prediction (i.e. label). Demand of probabilistic prediction has arisen in view of the fact that sometimes we appreciate probabilities more than single predictions. A simple example is the probabilistic weather forecasting.

But in some area, single probabilistic prediction has not yet been enough. The term *multi-probabilities* is then brought to mind, namely, we announce several probability distributions for the new label rather than a solitary one. Venn predictor (or Venn Machine) is one of the multi-probabilistic classification systems [8]. There are many Venn predictors, each taxonomy used in the algorithm defines a Venn predictor even if the underlying algorithms are the same.

In our previous paper [10], we introduced a Venn predictor with Support Vector Machines (SVM) as its underlying algorithm, which converts numerical predictions of SVM into a taxonomy. That approach was applicable to any method that initially supplied predictions with prediction scores such as the distance to the hyperplane in SVM. Nonetheless, the process is very simple: all available scores are firstly sorted and then divided into several groups by equal-length intervals according to which interval the score lies. Each of these groups is a category. However, that approach could only be applied in binary cases. In this paper, we propose a method to generalize binary Venn Machine with SVM to a method capable for multi-class cases. Then we consider two alternative methods that may be more accurate: SVM Venn Machine with k-means clustering and Venn-ABERS predictor. These two algorithms are also applicable to any machine learning algorithms with prediction scores.

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 251–260, 2014.

2 Methodology

In this section, two kinds of Venn predictors that use SVM as their underlying algorithm will be introduced together with our alternative methods. They are SVM Venn Machine with homogeneous intervals (VM-SVM-HI) generalized from the binary-only version of [10] together with our alternative method SVM Venn Machine with k-means clustering (VM-SVM-KM) and the Venn-ABERS predictor based on SVM (VA-SVM) proposed by Vladimir Vovk [7]. The former two algorithms could be implemented in both multi-class cases and binary cases, while VA-SVM could only deal with binary data sets.

2.1 Venn Machine

Venn Machine is a multi-probabilistic predictor described in [8]. The basic idea of Venn Machine is to divide every example into its corresponding category based on certain rules and then the frequencies of labels in the chosen category are used as probabilities for the new object's label. Taxonomy is the way how the examples are divided into categories. The underlying algorithm is the algorithm used in the taxonomy.

Assuming a standard machine learning classification problem: given a training set of examples $z_1, z_2, \ldots, z_{n-1}$. Each z_i consists of a pair of object x_i and label y_i. The possible labels y_i ($y_i \in \mathbf{Y}$) are finite. And we are also given a test object x_n. Our task is to predict the label y_n for the new object x_n and give the estimation of the likelihood that our prediction is correct.

Supposing we have a taxonomy A_n, consider a label $y \in \mathbf{Y}$ for the new object x_n. A_n assigns a category τ_i to an example z_i

$$\tau_i = A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i) \tag{1}$$

where n is the number of objects in the bag, $\tau_i \in \mathbf{T}$ is one of the finite categories and z_i is the pair (x_i, y_i), z_n is the pair (x_n, y).

Moreover, we assign z_i and z_j to the same category if and only if

$$A_n(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i) = A_n(\wr z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_n \int, z_j) \tag{2}$$

The category τ_n contains $z_n = (x_n, y)$. Let p_y be the empirical probability distribution of the labels in category τ_n.

$$p_y(y') := \frac{|\{(x^*, y^*) \in \tau_n : y^* = y'\}|}{|\tau_n|} \tag{3}$$

p_y is a probability distribution on $\mathbf{Y}$.

Having tried every possible label for x_n, we get a Venn predictor. The predictor $P_n := \{p_y : y \in \mathbf{Y}\}$ is a multi-probabilistic predictor consists of K distributions, where $K = |\mathbf{Y}|$. Then we could calculate a $K \times K$ frequency matrix P. The *quality* of a column is the minimum entry of the column. Let the *best* column which has the highest *quality* be j_{best}. Then our predicted label is j_{best} and the interval of possibility that our prediction is correct is

$$[\min_{i=1,\ldots,K} P_{i,j_{best}} \ , \ \max_{i=1,\ldots,K} P_{i,j_{best}}] \tag{4}$$

Underlying Algorithm for Taxonomy. Any algorithm that generates or predicts a numeric score for the example could be implemented in our taxonomy. However, we mainly focus on Venn predictors with SVM as the underlying algorithm in this paper. The decision function in SVM is a kind of scoring functions. Therefore, we use the values derived from the decision function of SVM (i.e. the values prior to applying a sign function) as part of our design.

Homogenous Intervals. One of the simplest ways to design taxonomies is stated as follows. Firstly we use the training set to train an SVM and calculate the decision values ($d(x) = \langle w, x_i \rangle + d$) of all examples in the training set and the new object. Secondly the whole range of decision values obtained will be divided into several intervals of equal length. Each interval is a category and objects of which the decision values fall into the same interval are of the same category. This design was introduced in [10] and could only used in binary case. Now we will discuss the generalization and alternative to it.

Combined Decision Function. In multi-class cases, we will have several binary SVM classifiers regardless of whether One-vs-One or One-vs-All approach is used. A scheme for multi-class SVM using One-vs-All approach was developed by Lambrou et al. in [5], which uses the largest decision value as the score. Generally, One-vs-One SVM is more efficient in accuracy than One-vs-All SVM. Therefore, we need to develop a new function to combine the outputs of all One-vs-One SVM classifiers and transform them into a single prediction score which could be used by Venn Machine. We call such function a *Combined Decision Function*.

For a data set with k possible labels: $\{0, 1, \ldots, k-1\}$, there are $k(k-1)/2$ binary SVM if we use One-vs-One approach. For each possible label, there are $k-1$ related SVM decision functions. Then we use (5) to calculate the combined decision function $D(x)$ for the new example x,

$$D(x) = \hat{y} + \frac{1}{k-1} \sum_{i=0, i \neq \hat{y}}^{k-1} N(f_{\hat{y}i}(x)) \tag{5}$$

where $\hat{y}$ is the overall predicted label done by max-wins voting strategy in One-vs-One SVM, $f_{\hat{y}i}(x)$ is the decision function of SVM classifier on $\hat{y}$-vs-i, N is a function that does the normalized transformation to $[0, 1]$. Another point we need to declare here is that in $f_{\hat{y}i}(x)$ we always put $\hat{y}$ before i which means we need to apply an opposite operation when $\hat{y}$ is greater than i. Since the examples of label $\hat{y}$ are treated as negative examples in i-vs-$\hat{y}$ classifier of a binary SVM.

This function firstly selects all $k-1$ related SVM and applies an opposite operation if $\hat{y}$ is not treated as the positive class in the binary SVM classifier. Then it does the normalisation to transform the values into $[0, 1]$. Finally, we output the arithmetic mean of them added with $\hat{y}$ as the combined decision value of new example x. The reason that adding $\hat{y}$ to the arithmetic mean is that it could prevent the decision values of different classes stack at the same area.

Dividing Intervals by *k*-Means Clustering. Instead of dividing the intervals homogeneously, we came up with a new dividing scheme, which uses k-means clustering [4, 6] to divide all decision values.

k-means clustering is a cluster analysis method which aims to divide n objects into k clusters in which each object belongs to the cluster with the nearest mean. Given a set of objects $(x_1, x_2, \ldots, x_n)$, where each object $x_i \in \mathbf{R}^d$ is a d-dimensional real vector, k-means clustering aims to partition the n objects into k sets $(k \leq n)$ $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimise the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{6}$$

where μ_i is the mean value of points in S_i.

In our design, dimension d is fixed to "1", while the number of clusters is equal to the number of possible labels. So the heuristic algorithm we used could be described as below.

1. k initial means values are randomly generated within the data domain.
2. k clusters are created (or reassigned) by associating every object with the nearest mean value.
3. The centroid of each of the k clusters becomes the new mean value.
4. Steps 2 and 3 are repeated until the change of WCSS (6) between two states declines to be less than $\epsilon = 10^{-4}$.

Having applied k-means clustering, we divided the decision values into categories which could be used to calculate the matrix for new examples and make the probabilistic predictions as the standard Venn Machine does.

2.2 Venn-ABERS Predictor

Venn-ABERS predictor is a recently developed algorithm for multi-probabilistic prediction. It is modified from Zadrozny and Elkan's procedure of probability forecasting [9], which cannot be well calibrated. The modification introduced Venn predictors into the procedure to overcome the problem of potentially weak calibration as a result of the fact that Venn predictors are always well calibrated and guaranteed to be well calibrated under the exchangeability assumption. The basic idea of pre-trained Venn-ABERS predictor is that the training set is split into two parts: the proper training set and the calibration set. The proper training set is used to train the learning machine and predict the label for new examples, while the calibration set is used to calculate the probabilistic outputs for the predicted labels. The calibration set will be turned into a monotonically increasing set in this algorithm according to [1].

Before we discuss Venn-ABERS predictor, there are some notions to be introduced yet. First notion is the term "*scoring algorithm*". Scoring algorithm is an algorithm that trains a classifier on the training set and uses the classifier to

output a prediction score $s(x)$ for the new example x and predicts the label of x to be "1" if and only if $s(x) \geq c$ (c is a fixed threshold). So s is hereby called the *scoring function*. Many machine learning algorithms for classification are scoring algorithms. In our case, as what SVM defines, the decision function of SVM is a scoring function, since we assign a new example the positive label "+1" if and only if its decision value is greater than zero and vice versa for the negative label. The second notion is "*isotonic calibrator*", which is a monotonically increasing function on the set $\{s(x_1), \ldots, s(x_l)\}$ that maximizes the likelihood

$$\prod_{i=1}^{l} p_i, \text{ where } p_i := \begin{cases} g(s(x_i)) & \text{if } y_i = 1 \\ 1 - g(s(x_i)) & \text{if } y_i = 0 \end{cases} \tag{7}$$

this function g is unique and can be found by using the "pool-adjacent violators algorithm" (PAVA) introduced in [1].

The workflow of Venn-ABERS predictor is as follows. Assuming a standard binary machine learning problem: a training set of examples $z_1, z_2, \ldots, z_l$. Each z_i consists of a pair of object x_i, and label y_i. The possible labels are binary, that is, $y \in \{0, 1\}$. And we are also given a test object x. Our task is to predict the label y for the new object x and give the estimation of the likelihood that our prediction is correct.

Let us split the training set $\wr z_1, z_2, \ldots, z_l \int$ into two parts: the proper training set $\wr z_1, z_2, \ldots, z_m \int$ of size m ($m < l$) and the calibration set $\wr z_{m+1}, z_{m+2}, \ldots, z_l \int$. And $s : \mathbf{X} \to \mathbb{R}$ is the scoring function of training set $\wr z_1, z_2, \ldots, z_m \int$. Given a new example x, we have two calibrators. Let g_0 be the isotonic calibrator for $\wr (s(x_{m+1}), y_{m+1}), (s(x_{m+2}), y_{m+2}), \ldots, (s(x_l), y_l), (s(x), 0) \int$, g_1 be the calibrator for $\wr (s(x_{m+1}), y_{m+1}), (s(x_{m+2}), y_{m+2}), \ldots, (s(x_l), y_l), (s(x), 1) \int$.

To achieve the isotonic calibrator, we do the followings according to the definition of PAVA. First we arrange the pairs $(s(x_i), y_i)$ in the increasing order according to the values of score function $s(x_i)$. Having obtained a binary sequence consisting of labels y_i, we applied PAVA to find the increasing sequence of them. The final isotonic calibrator g is a function mapping the increasing scores to the increasing sequence (i.e. probabilities). As the score increases, the object is more likely to be "1" in correlation with the increasing sequence.

Then the multi-probability prediction outputs for that the predicted label should be "1" is $\{p_0, p_1\}$, where $p_0 := g_0(s(x))$ and $p_1 := g_1(s(x))$. And for the reason that we need to predict the probability for the prediction label is correct, we should transform the bounds $\{p_0, p_1\}$ to $\{1 - p_1, 1 - p_0\}$ when the predicted label is "0".

3 Experimental Results

To compare our algorithm to SVM Venn Machine with homogeneous intervals and Venn-ABERS predictor, we used eight data sets from the real world which could be easily obtained from UCI Repository (`http://archive.ics.uci.edu/ml/`) except that SVMguide1 is obtained from the website of LibSVM [3]. The data sets we

Table 1. Main characteristics for each data set

Data Set	# of Objects	# of Features	# of Classes	Training Set Size	Testing Set Size
WBC	683	10	2	400	283
SVMguide1	7089	4	2	3089	4000
Splice	3175	60	2	1000	2175
Satimage	6435	36	6	4435	2000
Segment	2310	19	7	1500	810
DNA	3186	180	3	2000	1186
Wine	178	13	3	100	78
Vehicle	846	18	4	500	346

used in this paper could be divided into two parts based on their number of classes. The details of these data sets are summarised in Table 1.

3.1 Experimental Settings

For VM-SVM-KM, the number of clusters and the initial means, which are the two key features of k-means clustering, are often regarded as its biggest drawbacks. The number of clusters is an input parameter: an inappropriate choice of k may yield poor results. That is why, when performing k-means clustering, it is important to run diagnostic checks for determining the number of clusters in the data set. The choice of initial means might lead the convergence to a local minimum which may produce counterintuitive results. A good design of a combined decision function could make it easier to avoid these two drawbacks.

To have a more intuitive view of our combined decision function described in (5), we applied the algorithm to Satimage data set and plotted the histogram in Fig. 1, roughly representing the distribution of the decision values.

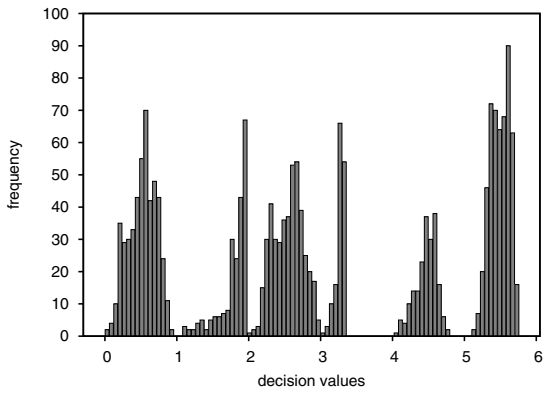


Fig. 1. Histogram of combined decision values for the Satimage data set

It can be seen obviously from the figure that there were 6 clusters in the data set, the exact number of the possible labels. The reason for this is that the decision function spreads out the values into $(0, k)$ by adding the most possible labels. Furthermore, each cluster i $(i = 1, 2, \ldots, k)$ is approximately within the range of $(i - 1, i)$, which means we could choose the initial means from each range to avoid the local minimum trap as much as possible and speed up the convergence process. We conducted k-means clustering to these decision values and calculated the 6 centroids: $0.63, 1.91, 2.64, 3.33, 4.57, 5.59$. The result seems to be a reasonable reflection of the histogram.

Then we could come to our decision that we set the number of clusters the same as the number of possible labels and we choose the initial means as $0.5, 1.5, \ldots, k - 0.5$ if the possible labels are $0, 1, \ldots, k - 1$.

Additionally, we need to notice that k-means clustering uses Euclidean distance as a metric and variance as a measure of cluster scatter, which makes it tend to produce equal-sized clusters. Since data is split halfway between cluster means, this can lead to suboptimal splits as some objects will be attributed to the incorrect cluster, especially for unbalanced data set as Satimage data set.

Except all the settings for the underlying algorithm, we need another setting for VA-SVM. It is the size of the calibration set. Having given careful consideration to both accuracy and narrowness of the bounds, we decided to take 30% of the whole data set as the calibration set, And the calibration set was stratified selected from the whole training set, which means the distribution of classes in the calibration set was the same as in the training set.

Although the size of proper training set in VA-SVM is smaller comparing to the size of training set in our algorithms, this is still a fair comparison because we use the same original training set for all algorithms, otherwise VA-SVM will need extra examples for probabilistic predictions. We also noticed that Venn-ABERS predictor is an inductive Venn predictor while Venn Machine is a transductive Venn predictor. The gap between inductive and transductive learning algorithms are not distinguishable in our offline setting. Because in offline setting, we use the fixed predictors to make predictions for testing set. Furthermore, in VA-SVM we repeat the computations of isotonic calibrators for each testing object which still involve all examples in calibration set.

3.2 Comparisons and Results

For binary cases, we applied VM-SVM-KM, VM-SVM-HI and VA-SVM to the data sets in the offline setting. While for multi-class cases, we only applied VM-SVM-KM and VM-SVM-HI to the data sets in both offline setting and online setting. Hence, there were three comparisons described as below. All the SVMs in these algorithms were using RBF kernel. Additionally, the parameters of SVM for each data set, including cost C and σ in RBF kernel, were determined by grid

search on the training set and retained the same over corresponding algorithms respectively. The algorithms were compared in terms of their accuracies and probabilistic outputs in these data sets. In addition, we calculated the Brier scores (introduced in [2]) of the mean of the probabilistic bounds as evaluation for binary data sets.

The experimental results of VM-SVM-KM compared with VM-SVM-HI and VA-SVM are shown in Table 2.

Table 2. The offline accuracy and probability results on the binary data set

Data Set	Taxonomy	Accuracy	Prob. Outputs	Brier Score
WBC	VM-SVM-KM	97.53%	[86.34%,98.94%]	0.0325
	VM-SVM-HI	97.22%	[83.63%,98.70%]	0.0369
	VA-SVM	97.17%	[85.67%,95.97%]	0.0315
SVMguide1	VM-SVM-KM	96.93%	[91.27%,98.42%]	0.0362
	VM-SVM-HI	95.79%	[89.59%,98.97%]	0.0406
	VA-SVM	95.95%	[93.67%,96.29%]	0.0370
Splice	VM-SVM-KM	90.21%	[82.44%,96.07%]	0.0884
	VM-SVM-HI	89.52%	[80.15%,97.35%]	0.0939
	VA-SVM	89.15%	[83.40%,88.32%]	0.0878

The comparison results of our algorithm against VM-SVM-HI for all multi-class data sets in the offline setting are shown in Table 3.

Table 3. The offline accuracy and probability results on the multi-class data set

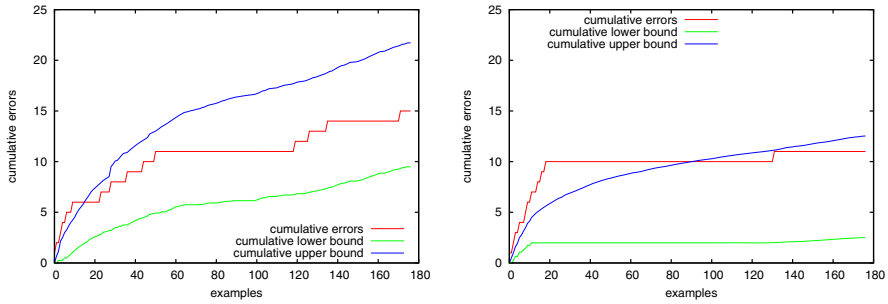
Data Set	Algorithm	Accuracy	Probabilistic Outputs
Satimage	VM-SVM-HI	84.18%	[75.48%,96.92%]
	VM-SVM-KM	86.56%	[81.18%,93.33%]
Segment	VM-SVM-HI	90.88%	[74.61%,96.68%]
	VM-SVM-KM	91.65%	[75.64%,95.60%]
DNA	VM-SVM-HI	94.34%	[81.39%,98.07%]
	VM-SVM-KM	96.65%	[87.25%,99.48%]
Wine	VM-SVM-HI	92.30%	[81.19%,97.11%]
	VM-SVM-KM	96.11%	[86.53%,98.47%]
Vehicle	VM-SVM-HI	67.63%	[56.42%,77.48%]
	VM-SVM-KM	69.15%	[60.65%,79.20%]

And the results for the online setting are shown in Table 4.

In order to giving a more intuitive comparison, we also give figures on online performance for Wine data set in Fig. 2.

Table 4. The online accuracy and probability results on the multi-class data set

Data Set	Algorithm	Accuracy	Probabilistic Outputs
Satimage	VM-SVM-HI	80.94%	[80.20%,81.69%]
	VM-SVM-KM	83.40%	[83.24%,83.86%]
Segment	VM-SVM-HI	88.40%	[88.92%,93.20%]
	VM-SVM-KM	89.96%	[90.11%,91.50%]
DNA	VM-SVM-HI	89.12%	[88.46%,89.86%]
	VM-SVM-KM	89.70%	[89.25%,90.48%]
Wine	VM-SVM-HI	91.53%	[87.57%,94.35%]
	VM-SVM-KM	93.22%	[91.67%,96.87%]
Vehicle	VM-SVM-HI	66.04%	[70.24%,72.17%]
	VM-SVM-KM	67.83%	[69.48%,71.02%]



(a) VM-SVM-HI (b) VM-SVM-KM

Fig. 2. Comparison of online performances for the Wine data set

4 Discussion and Conclusion

From the results shown in Table 2 which comparing our method to VM-SVM-HI and VA-SVM, we could draw the following conclusions.

First, VM-SVM-KM performed better in accuracy among these three data sets nevertheless the increases were small. Furthermore, VA-SVM used 30% of the training set as the calibration set which did not participate in the training of classifiers; hence it may lead to worse results. Second, the accuracies of VA-SVM slightly outnumbered the upper bound in WBC and Splice data sets, which could be due to the offline setting. Third, the probability bounds of VA-SVM were the narrowest while VM-SVM-HI had the widest bounds and VM-SVM-KM was in-between. This is the advantage of VA-SVM in view of our preference for narrow bounds. It is also backed by the Brier scores results: VA-SVM and VM-SVM-KM had close Brier scores while VM-SVM-HI had the worst results.

Another point is that VA-SVM does not calibrate their predicted label according to the probability, more specifically it is an algorithm that generates the probabilities from the scores only, while our algorithm gives predictions based on the highest likelihood. Except the improvement in accuracy, VM-SVM-KM is easy to configure because the number of clusters is the only input parameter of this algorithm which is equal to the number of classes.

From the results presented in Table 3 and Table 4 where the performance of VM-SVM-KM is compared with VM-SVM-HI in both offline and online setting, we can discover the following points.

First, it can be observed that all accuracies were within the probabilistic outputs in the offline setting, while in the online setting the accuracies exceeded the bounds in Segment and Vehicle data sets. Second, after implementing the k-means clustering, the accuracies are improved in both settings. However, in the offline setting, the improvements ranged from 0.8% to 3.8% depending on the data sets. In the online setting, the difference between these two algorithms became smaller, only 0.6% to 2.5%. Third, probability bounds become narrower after applying the k-means clustering, mostly benefiting from the rise of lower bounds. An intuitive comparison is shown in Fig. 2. The cumulative errors and cumulative error bounds in the figures all decreased after implementing k-means clustering, and the bounds became narrower in the meantime.

In summary, the improvement in each of the eight data sets was not significant which is due to the consistency of these data sets. Nevertheless, we still believe that SVM Venn Machine with k-means clustering is better when compared with homogeneous intervals since it could yield better accuracy and narrower bounds. However, in comparison with Venn-ABERS predictor, our algorithm is good on accuracy and weak on narrowness of the bounds. Despite that, our algorithm is easier to set up, and it predicts the most likely label while Venn-ABERS predictor only generates the probabilities.

References

1. Ayer, M., Brunk, H.D., Ewing, G.M., Reid, W.T., Silverman, E.: An empirical distribution function for sampling with incomplete information. Ann. Math. Statist. 26(4), 641–647 (1955)
2. Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly Weather Review 78(1), 1–3 (1950)
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
4. Forgy, E.W.: Cluster analysis of multivariate data: Efficiency vs interpretability of classifications. Biometrics 21, 768–769 (1965)
5. Lambrou, A., Papadopoulos, H., Nouretdinov, I., Gammerman, A.: Reliable probability estimates based on support vector machines for large multiclass datasets. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 182–191. Springer, Heidelberg (2012)
6. Lloyd, S.P.: Least squares quantization in pcm. IEEE Transactions on Information Theory 28, 129–137 (1982)
7. Vovk, V.: Venn predictors and isotonic regression. CoRR abs/1211.0025 (2012)
8. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer-Verlag New York, Inc., Secaucus (2005)
9. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 694–699. ACM (2002)
10. Zhou, C., Nouretdinov, I., Luo, Z., Adamskiy, D., Randell, L., Coldham, N., Gammerman, A.: A comparison of venn machine with platt's method in probabilistic outputs. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) EANN/AIAI 2011, Part II. IFIP AICT, vol. 364, pp. 483–490. Springer, Heidelberg (2011)

Efficiency Comparison of Unstable Transductive and Inductive Conformal Classifiers

Henrik Linusson[1], Ulf Johansson[1], Henrik Boström[2], and Tuve Löfström[1]

[1] School of Business and IT
University of Borås, Borås, Sweden
{henrik.linusson,ulf.johansson,tuve.lofstrom}@hb.se
[2] Dept. of Computer and Systems Sciences
Stockholm University, Kista, Sweden
henrik.bostrom@dsv.su.se

Abstract. In the conformal prediction literature, it appears axiomatic that transductive conformal classifiers possess a higher predictive efficiency than inductive conformal classifiers, however, this depends on whether or not the nonconformity function tends to overfit misclassified test examples. With the conformal prediction framework's increasing popularity, it thus becomes necessary to clarify the settings in which this claim holds true. In this paper, the efficiency of transductive conformal classifiers based on decision tree, random forest and support vector machine classification models is compared to the efficiency of corresponding inductive conformal classifiers. The results show that the efficiency of conformal classifiers based on standard decision trees or random forests is substantially improved when used in the inductive mode, while conformal classifiers based on support vector machines are more efficient in the transductive mode. In addition, an analysis is presented that discusses the effects of calibration set size on inductive conformal classifier efficiency.

1 Introduction

Conformal Prediction [1] is a machine learning framework for associating predictions for novel data with a measure of their *confidence*; whereas traditional machine learning algorithms produce *point predictions* — a single label $\hat{y}$ per test example — conformal predictors produce *prediction regions* — prediction sets $\hat{Y} \subseteq Y$ that, in the long run, contain the true labels for the test set with some predefined probability $1 - \epsilon$. Historically, such confidence predictions have relied on the Bayesian learning and Probably Approximately Correct (PAC) learning frameworks; however, the validity of Bayesian confidence predictions relies on an assumption of the *a priori* distribution, while PAC learning confidence measures apply to the entire model, and not the individual predictions [2]. In contrast, conformal predictors are able to produce confidence measures tailored for each separate prediction on novel data, and rely only on the assumption that the data is *exchangeable* — that the ordering of data points does not affect their joint

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 261–270, 2014.

distributions — which is an even weaker assumption than the i.i.d. assumption typically required by traditional machine learning algorithms.

The method of constructing prediction regions using conformal predictors relies on measuring the strangeness — the *nonconformity* — of each data point, using some (arbitrary) real-valued function, called a *nonconformity function*, that measures the degree of strangeness of an example (x_i, y_i) in relation to a bag (multiset) of examples $Z = \{(x_1, y_1), ..., (x_i, y_i), ..., (x_k, y_k)\}$. For classification problems, this nonconformity measure is often based on the predictions of a traditional machine learning algorithm, called the *underlying model* of the conformal predictor. By comparing the nonconformity of a tentative classification $(x_{k+1}, \tilde{y})$ for a novel (test) input pattern x_{k+1} to the nonconformity scores of previously seen data, a conformal predictor can make inferences as to whether $\tilde{y}$ is likely to be a correct classification for x_{k+1}, and thus decide whether or not to include $\tilde{y}$ in the prediction region $\hat{Y}_{k+1}$.

There are two major categories of conformal predictors: transductive conformal predictors (TCP) [3,4] and inductive conformal predictors (ICP) [2,5]; both variants are based on the same principles, and place the same exchangability requirement on the data. Where they differ is in their usage of the training data, and their overall training scheme. For a novel input pattern x_{k+1} and every tentative prediction $\tilde{y} \in Y$, TCP measures the nonconformity of all examples in the bag $Z' = Z \cup \{(x_{k+1}, \tilde{y})\}$ relative to each other, meaning that the instance $(x_{k+1}, \tilde{y})$ is considered when calculating the nonconformity scores for the training set Z. Hence, for every new pattern x_{k+n} and every $\tilde{y}$, the underlying model needs to be retrained, and the nonconformity scores for the training data recomputed, rendering TCP computationally intensive. In ICP, only part of the data is used to train the underlying model (once), while the remaining data (a *calibration set*) is set aside to provide an unbiased estimate of the distribution of nonconformity scores.

The predictive *efficiency* — that is, the size of the prediction regions produced — of any conformal predictor is closely tied to the nonconformity function's ability to rank examples by order of strangeness. Moreover, as noted in several papers, ICP models typically suffer a small loss in predictive efficiency compared to corresponding TCP models due to the reduced number of training and calibration examples [1, 5–8]. However, as pointed out in [1], an *unstable* nonconformity function — one that is heavily influenced by an outlier example, i.e., an erroneously labled test instance $(x_{k+1}, \tilde{y})$ — can cause TCP models to become inefficient.

So, the choice between TCP and ICP is not so clear-cut: on the one hand, ICP will surely always produce its predictions faster than TCP, and TCP is often expected to have a higher predictive power than ICP. On the other hand, the efficiency of TCP relies on the chosen nonconformity function being *stable*, meaning that the underlying model does not train outlier examples into its learned rule [1]. When choosing a conformal prediction setup, a user should thus consider not only the trade-off between predictive speed and predictive power in

TCP and ICP, but also whether the chosen nonconformity function can be used effectively in TCP.

Decision trees and random forests are commonly used model types that are able to accommodate their training examples well, and, due to their ability to near-perfectly fit their training data, these model types should be expected to function better as underlying models in ICP than in TCP. To the best of our knowledge, however, this claim has not been investigated in existing literature.

In this study, the efficiency of TCP classifiers based on off-the-shelf decision trees and random forests, implemented by the scikit-learn [9] Python package, is compared to the efficiency of corresponding ICP classifiers; the results of this comparison are juxtaposted with an identical comparison of TCP and ICP classifiers based on stable support vector machines. To allow for a straightforward comparison of TCP and ICP classifiers, an analysis is also presented that discussess the effects of calibration set size on ICP classifier efficiency.

2 Background

Given some nonconformity scoring function $\Delta(Z, x_i, y_i) \rightarrow \overline{\mathbb{R}}$, a conformal predictor can produce prediction regions that are *valid* in the sense that they contain the true target with some predefined probability $1 - \epsilon$. For classification problems, a common approach is to define a nonconformity function by combining a traditional machine learning algorithm and some error measure of the algorithm's predictions, e.g., the margin scoring function used in [10]:

$$\Delta(h, x_i, y_i) = P(y_i \mid h(x_i)) - \arg\max_{y' \neq y_i} P(y' \mid h(x_i)), \tag{1}$$

where h is a predictive model trained on some set Z, i.e., h represents a generalization of Z. Given the nature of classification problems in general, it is to be expected that a test pattern that deviates from the examples found in Z is likely to be misclassified by h, i.e., examples that are not conforming to Z are likely to be assigned large nonconformity scores.

When making a prediction using a conformal classifier, the nonconformity function is applied to a set of examples with known labels, resulting in a set of nonconformity scores $\alpha_1, ..., \alpha_k$ that represents a sample which is to be used for statistical inference (we here refer to this as the *nonconformity baseline*). The test pattern x_{k+1} is then assigned a tentative classification $(x_{k+1}, \tilde{y})$, where $\tilde{y} \in Y$, and a nonconformity score $\alpha_{k+1} = \Delta(Z, x_{k+1}, \tilde{y})$. Using a form of hypothesis testing (described in Sections 2.1 and 2.2), the conformal classifier attempts to reject the null hypothesis that $(x_{k+1}, \tilde{y})$ is conforming with Z, i.e., if the nonconformity score α_{k+1} is higher than for most examples in Z, as estimated by the nonconformity baseline, then $\tilde{y}$ is considered to be an unlikely classification for x_{k+1} and can be excluded from the final prediction set.

2.1 Transductive Conformal Predictors

A transductive conformal classifier uses the full training set Z to establish the nonconformity baseline. Due to the exchangeability assumption, this requires that the test pattern $(x_{k+1}, \tilde{y})$ is considered when calculating the nonconformity scores $\alpha_1, ..., \alpha_k$, i.e., if the nonconformity function Δ is based on an inductive model h, then $(x_{k+1}, \tilde{y})$ must be included in the training set for h. If it is not, then there is a possibility that h will have a larger bias towards its training examples than towards $(x_{k+1}, \tilde{y})$, meaning that the training examples and the test pattern might have different expected nonconformity values (which is a direct violation of the exchangeability assumption[1]). TCP thus requires that the underlying model h is trained $n\,|Y|$ times, where n is the number of test patterns, using the following scheme:

1. Assume a label $\tilde{y}$ and form the set $Z' = Z \cup \{(x_{k+1}, \tilde{y})\}$.
2. Use Z' to train a model h.
3. For each $(x_i, y_i) \in Z'$ calculate $\alpha_i = \Delta(h, x_i, y_i)$.
4. Calculate the p-value for $(x_{k+1}, \tilde{y})$ as

$$p(x_{k+1}, \tilde{y}) = \frac{|\{z_i \in Z' \mid \alpha_i \geq \alpha_{k+1}\}|}{|Z'|}\,. \tag{2}$$

5. If $p(x_{k+1}, \tilde{y}) > \epsilon$, include $\tilde{y}$ in the prediction region $\hat{Y}_{k+1}$.

2.2 Inductive Conformal Predictors

Inductive conformal predictors instead use only part of the training data to fit the underlying model h, setting aside a calibration set that is later used to establish the nonconformity baseline. Since the proper training set $Z^t \subset Z$ used to fit h and the calibration set $Z^c \subset Z$ are disjoint, the nonconformity scores $\alpha_1, ..., \alpha_{k-t}$ are exchangeable (unbiased) with the nonconformity score α_{k+1} without the need for including $(x_{k+1}, \tilde{y})$ in the training of h; thus, an ICP only needs to be trained once:

1. Divide Z into two disjoint subsets Z^t and Z^c.
2. Use Z^t to train a model h.
3. For each $(x_i, y_i) \in Z^c$, calculate $\alpha_i = \Delta(h, x_i, y_i)$.

For a novel test instance x_{k+1}:

1. Assume a label $\tilde{y}$ and calculate $\alpha_{k+1} = \Delta(h, x_{k+1}, \tilde{y})$.
2. Calculate the p-value for $(x_{k+1}, \tilde{y})$ as

$$p(x_{k+1}, \tilde{y}) = \frac{|\{z_i \in Z^c \mid \alpha_i \geq \alpha_{k+1}\}| + 1}{|Z^c| + 1}\,. \tag{3}$$

3. If $p(x_{k+1}, \tilde{y}) > \epsilon$, include $\tilde{y}$ in the prediction region $\hat{Y}_{k+1}$.

[1] If the model is able to better predict the correct output for the training examples than the test example, then the p-values for the true targets will no longer be uniformly distributed as required by the conformal prediction framework [1].

3 Method

The main point of interest of this study is the efficiency comparison of TCP and ICP classifiers based on decision tree, random forest and support vector machine models, but, to provide a straightforward comparison between the two, it is also necessary to find a suitable choice of calibration set size for the ICP classifiers. To the best of our knowledge, no thorough investigation has been published that discusses the effects of calibration set size on ICP classifier efficiency, and so the results presented in this paper contain first a discussion of ICP classifier setup, and second a comparison of ICP and TCP classifier efficiency.

To investigate what effect the calibration set size has on the efficiency of ICP classifiers using various types of underlying models, several ICP models were trained on five binary classification sets from the LIBSVM website [11] (a9a, covertype, cod-rna, ijcnn1 and w8a), using different amounts of training and calibration data. For each dataset, stratified random samples of $s = 500, 1000, ..., 4500$ examples were drawn, and for each s, several ICP models were applied to the same test set of 100 examples, each ICP model using $c = 100, 200, ..., s - 100$ calibration examples and $t = s - c$ training examples.

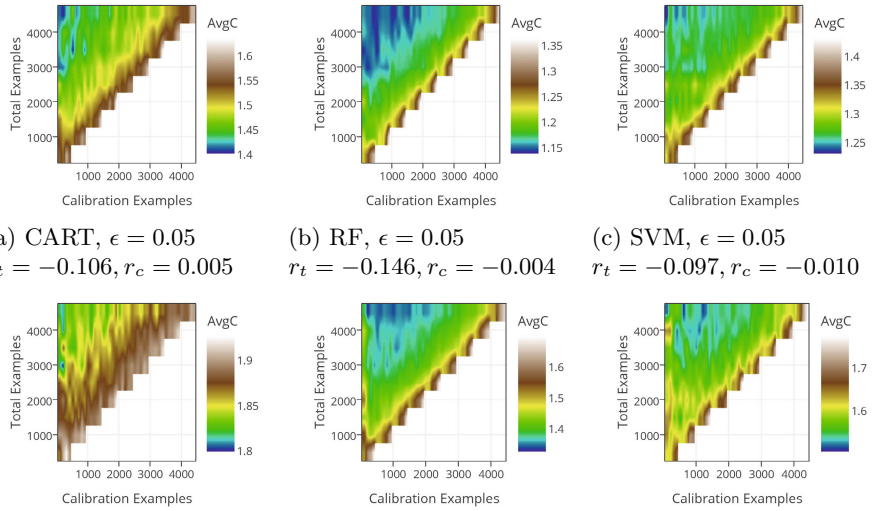
To compare the efficiency of the ICP and TCP variants, both types of classifiers were trained on 19 binary classification sets from the UCI repository [12]. The ICP models used a suitable calibration size as suggested by the results from the first experiment (20%). To limit the training time required for the TCP models, results from the larger datasets (kr-vs-kp, sick and spambase) were obtained using 1x10-fold cross-validation, while the results from the remaining sets were obtained using 5x10-fold cross-validation.

In both experiments, three different types of underlying models were used: decision trees (CART) [13], random forests (RF) [14] and support vector machines (SVM) [15]. The CART models used relative frequency probability estimates and no pruning, making them highly susceptible to overfitting noisy data. RF models (here consisting of 100 trees) are relatively robust to noise, but since the ensemble members are simple CART models, noisy examples are still fit into the learned rule to some extent. SVM models are, in contrast, stable to isolated noisy data points [16]. The scikit-learn library [9] for Python, using default settings, was used to train the underlying models. The margin nonconformity function (1) was used to construct the conformal classifiers, and predictions were made in the off-line (batch) mode.

4 Results

Figure 1 shows the relationships between training set size, calibration set size and efficiency (measured as AvgC — the average number of labels per prediction). The size of the calibration set is expressed as a portion of the full dataset, i.e., the point $(500, 1000)$ uses 500 of the total (1000) available examples as calibration data, and the remaining 500 examples as training data. The results are averaged over all five datasets and all five iterations. Clearly, the more data that is made

available to the ICP classifier, the more efficient its predictions are, but only if the data is used sensibly. If more data is made available, and all newly available examples are added to the calibration set (any line $y = x+k$ in the six plots), the efficiency remains virtually unchanged at $\epsilon = 0.05$. If instead all newly available examples are added to the proper training set (any line parallel to the Y axis) the efficiency of the ICP model increases greatly, whereas, naturally then, the efficiency decreases if a larger calibration set is used while the total amount of available data remains unchanged (any line parallel to the X axis). The chosen (absolute) size of the calibration set thus has a small effect on ICP classifier efficiency, while the size of the proper training set is more strongly correlated with efficiency (cf. the correlations, r_t for training set size and r_c for calibration set size, listed below the plots).



(a) CART, $\epsilon = 0.05$
$r_t = -0.106, r_c = 0.005$

(b) RF, $\epsilon = 0.05$
$r_t = -0.146, r_c = -0.004$

(c) SVM, $\epsilon = 0.05$
$r_t = -0.097, r_c = -0.010$

(d) CART, $\epsilon = 0.01$
$r_t = -0.135, r_c = -0.001$

(e) RF, $\epsilon = 0.01$
$r_t = -0.205, r_c = -0.023$

(f) SVM, $\epsilon = 0.01$
$r_t = -0.103, r_c = -0.010$

Fig. 1. ICP AvgC based on portion of examples used in the calibration set. The y-axis represents a growing dataset, while the x-axis represents a growing calibration set (relative to the full dataset). Blue areas represent the most efficient combinations of training and calibration set sizes, green and yellow areas are moderately efficient, while brown areas are the most inefficient.

As illustrated by Figures 1d, 1e and 1f, the size of the calibration set plays a larger role in determining efficiency when $\epsilon = 0.01$ than it does when $\epsilon = 0.05$, although the training set size is clearly the most important factor for maximizing efficiency at both significance levels. The best performing ICP classifiers are those using $15 - 30\%$ of the full training set as their calibration set. Notably, the performance of ICP classifiers using a calibration set containing less than 500

examples is quite poor when $\epsilon = 0.01$, regardless of training set size, suggesting that at least a few hundred examples should be used for calibration unless this leaves too few examples in the proper training set. The coverage (i.e. the 'empirical validity') of the conformal classifiers is not tabulated, however, all tested conformal classifiers show a coverage at or near the expected error rates on average. Note though, that the conformal classifiers using a very small calibration set (100 examples) displayed a larger variance in their coverage than the conformal classifiers using 500 calibration examples; in turn, using 500 calibration examples showed no substantial increase in variance compared to using a larger calibration set.

Table 1. AvgC of TCP and ICP based on CART, RF and SVM, $\epsilon = 0.05$

			CART		RF		SVM	
dataset	#f	#ex	ICP	TCP	ICP	TCP	ICP	TCP
balance-scale	4	577	1.571	1.891	1.037	1.044	0.957	0.951
breast-cancer	9	286	1.878	1.859	1.719	1.809	1.711	1.733
breast-w	10	699	1.259	1.520	0.973	0.976	0.987	0.993
credit-a	15	690	1.747	1.902	1.284	1.328	1.766	1.774
credit-g	20	1000	1.845	1.902	1.548	1.596	1.821	1.849
diabetes	20	769	1.842	1.896	1.541	1.601	1.822	1.861
haberman	3	306	1.849	1.843	1.649	1.773	1.680	1.607
heart-c	14	303	1.796	1.903	1.402	1.452	1.885	1.849
heart-h	14	264	1.808	1.898	1.402	1.460	1.838	1.836
heart-s	13	270	1.813	1.896	1.445	1.460	1.904	1.900
hepatitis	19	155	1.821	1.885	1.400	1.350	1.844	1.820
ionosphere	34	351	1.588	1.896	1.057	1.047	1.059	1.040
kr-vs-kp	36	3196	0.949	1.895	0.953	0.949	1.036	1.016
labor	16	57	1.719	1.909	1.400	1.115	1.525	1.039
liver-disorders	7	345	1.859	1.880	1.753	1.740	1.857	1.846
sonar	30	208	1.829	1.904	1.373	1.401	1.720	1.648
tic-tac-toe	9	958	1.255	1.906	0.960	0.963	1.161	1.045
sick	30	3772	0.964	1.886	0.954	0.950	1.062	1.036
spambase	57	4601	1.370	1.549	1.021	1.028	1.050	1.052
mean rank			**1.105**	1.895	1.316	1.684	1.684	1.316

Tables 1 and 2 show a comparison of the efficiency of ICP and TCP classifiers based on CART, random forest and SVM models. Note that the ranks are computed only within each TCP-ICP pair, i.e., the different underlying model types are not compared to each other in terms of efficiency ranks. Mean ranks in bold indicate significant differences at $\alpha = 0.05$ as reported by a two-tailed Wilcoxon signed-ranks test.

The stable SVM models do indeed appear to gain from being able to use the full dataset for both training and calibration when used in the TCP mode. At $\epsilon = 0.01$, the SVM-based TCP classifiers are significantly more efficient that SVM-based ICP classifiers; at $\epsilon = 0.05$ the SVM-TCP classifiers are more efficient on

Table 2. AvgC of TCP and ICP based on CART, RF and SVM, $\epsilon = 0.01$

			CART		RF		SVM	
dataset	#f	#ex	ICP	TCP	ICP	TCP	ICP	TCP
balance-scale	4	577	1.911	1.979	1.175	1.217	1.039	1.022
breast-cancer	9	286	1.977	1.976	1.931	1.965	1.940	1.972
breast-w	10	699	1.843	1.589	1.178	1.187	1.256	1.228
credit-a	15	690	1.951	1.982	1.806	1.810	1.934	1.936
credit-g	20	1000	1.969	1.981	1.838	1.871	1.959	1.972
diabetes	20	769	1.970	1.978	1.798	1.875	1.964	1.979
haberman	3	306	1.971	1.959	1.894	1.936	1.935	1.922
heart-c	14	303	1.966	1.981	1.800	1.815	1.982	1.976
heart-h	14	294	1.963	1.980	1.814	1.821	1.950	1.933
heart-s	13	270	1.953	1.979	1.837	1.827	1.984	1.979
hepatitis	19	155	1.960	1.983	1.832	1.758	1.976	1.966
ionosphere	34	351	1.917	1.974	1.589	1.454	1.543	1.330
kr-vs-kp	36	3196	1.067	1.980	1.008	1.000	1.203	1.258
labor	16	57	1.952	1.987	1.845	1.683	1.914	1.615
liver-disorders	7	345	1.978	1.959	1.945	1.929	1.969	1.948
sonar	30	208	1.968	1.980	1.804	1.672	1.922	1.872
tic-tac-toe	9	958	1.828	1.978	1.059	1.062	1.488	1.328
sick	30	3772	1.356	1.957	1.013	1.021	1.749	1.621
spambase	57	4601	1.875	1.673	1.272	1.623	1.352	1.344
mean rank			**1.263**	1.737	1.368	1.632	1.737	**1.263**

average, although the difference is not significant. It is evident, however, that the unpruned CART models — the most noise sensitive of the three model types — struggle with rejecting erroneous class labels in a TCP setting, likely due to the noisy test examples $(x_i, \tilde{y} \neq y_i)$ being fitted into the learned rules. TCP models based on CART are significantly less efficient than ICP-CART, both at $\epsilon = 0.05$ and $\epsilon = 0.01$. As expected, the RF models fare better in terms of TCP efficiency; the overfitting of the underlying CART models is counteracted by the smoothing effect of the ensemble constellation, however, not to such an extent that the TCP-RF models are more efficient than ICP-RF. ICP-RF is more efficient than TCP-RF on average, but the difference is not significant, neither at $\epsilon = 0.05$ nor at $\epsilon = 0.01$.

5 Related Work

As noted in [1], TCP classifiers using unstable nonconformity functions can be made more efficient through a slight modification of the transductive procedure. This modification involves calculating the nonconformity scores in a leave-one-out manner (LOOTCP), where the nonconformity of an example $(x_i, y_i) \in Z'$ is calculated from the set $Z' \setminus (x_i, y_i)$. Hence, in LOOTCP, the underlying model needs to be retrained not only for every test example and every tentative classification, but also for every training example. In principle, LOOTCP should

indeed be expected to produce more efficient predictions than ICP, however, as LOOTCP increases the computational complexity of the (already computationally intensive) TCP classifier by a factor k, it is questionable whether it is truly applicable in practice when the nonconformity function requires training.

In [17], random forests are used to construct conformal classifiers run in the transductive mode. Rather than using the entire forest when calculating the nonconformity scores for calibration and test examples, the error of the out-of-bag prediction is used, i.e., to calculate the nonconformity score for a specific example, only ensemble members not trained on that particular example are considered (similar to LOOTCP). Although not explicitly stated by the authors, this effectively results in a random forest TCP that is not affected by the potential issues of strange examples being trained into the model.

In [18], a method for estimating the prediction certainty in decision trees, similar to conformal prediction, is proposed. The authors note that the default tree induction algorithm requires a modification for the predictions to be useful. Since the goal is to identify examples that are difficult to predict, the test instance (which is, as in TCP, included in the training data) is only allowed to affect node splitting to a limited extent, to avoid fitting strange examples into the models.

6 Concluding Remarks

This study shows that inductive conformal classifiers based on standard decision tree and random forest models can be more efficient than corresponding transductive conformal classifiers, while the opposite is true for support vector machines. This is contrary to the commonly accepted claim that transductive conformal predictors are by default more efficient than inductive conformal predictors. It has also been shown that to maximize the efficiency of inductive conformal classifiers, the calibration set should be kept small relative to the amount of available data ($15 - 30\%$). At the same time, if the training set is large enough, it appears favourable to let the calibration set contain at least a few hundred examples.

For future work, we suggest that transductive and inductive conformal classifiers based on other types of classification models should be compared, to provide guidelines for designing conformal classification systems. Similarly, the efficiency of specialized transductive models, such as those proposed in [17], should be contrasted to the efficiency of standard transductive and inductive variants.

Acknowledgements. This work was supported by the Swedish Foundation for Strategic Research through the project High-Performance Data Mining for Drug Effect Detection (IIS11-0053) and the Knowledge Foundation through the project Big Data Analytics by Online Ensemble Learning (20120192).

References

1. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Verlag, DE (2006)
2. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. Tools in Artificial Intelligence 18, 315–330 (2008)
3. Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 148–155. Morgan Kaufmann Publishers Inc. (1998)
4. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 1999), vol. 2, pp. 722–726 (1999)
5. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive confidence machines for regression. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 345–356. Springer, Heidelberg (2002)
6. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. Tools in Artificial Intelligence 18(315-330), 2 (2008)
7. Papadopoulos, H., Vovk, V., Gammerman, A.: Conformal prediction with neural networks. In: 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, vol. 2, pp. 388–395. IEEE (2007)
8. Balasubramanian, V.N., Ho, S.S., Vovk, V.: Conformal prediction for reliable machine learning: theory, adaptations, and applications. Elsevier, Waltham (2013) (to appear)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research 12, 2825–2830 (2011)
10. Johansson, U., Boström, H., Löfström, T.: Conformal prediction using decision trees. In: IEEE International Conference on Data Mining (2013)
11. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), Software, available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
12. Bache, K., Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
13. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
14. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
15. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
16. Buciu, I., Kotropoulos, C., Pitas, I.: Demonstrating the stability of support vector machines for classification. Signal Processing 86(9), 2364–2380 (2006)
17. Devetyarov, D., Nouretdinov, I.: Prediction with confidence based on a random forest classifier. In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds.) AIAI 2010. IFIP AICT, vol. 339, pp. 37–44. Springer, Heidelberg (2010)
18. Costa, E.P., Verwer, S., Blockeel, H.: Estimating prediction certainty in decision trees. In: Tucker, A., Höppner, F., Siebes, A., Swift, S. (eds.) IDA 2013. LNCS, vol. 8207, pp. 138–149. Springer, Heidelberg (2013)

Anomaly Detection of Trajectories with Kernel Density Estimation by Conformal Prediction

James Smith[1], Ilia Nouretdinov[1], Rachel Craddock[2], Charles Offer[2], and Alexander Gammerman[1]

[1] Computer Learning Research Center, Royal Holloway University of London
{James.Smith.2009,alex,ilia}@cs.rhul.ac.uk
[2] Thales UK
{firstname.lastname@uk.thalesgroup.com}

Abstract. This paper describes conformal prediction techniques for detecting anomalous trajectories in the maritime domain. The data used in experiments were obtained from Automatic Identification System (AIS) broadcasts – a system for tracking vessel locations. A dimensionality reduction package is used and a kernel density estimation function as a non-conformity measure has been applied to detect anomalies. We propose average p-value as an efficiency criteria for conformal anomaly detection. A comparison with a k-nearest neighbours non-conformity measure is presented and the results are discussed.

1 Introduction

Anomaly detection is a large area of research in machine learning and many interesting techniques have been developed to detect 'abnormal' behaviour of objects. The word 'anomaly' here is used in the sense that there are some patterns in the data that do not conform to typical behaviour. These non-conforming patterns are often called 'anomalies' or 'abnormalities' or 'outliers' [1]. Recently some new techniques known as conformal predictors (CP) have emerged which allow the detection of the non-conformal behaviour of objects using some measures of non-conformity [2,3]. This technique also has an advantage in delivering provably valid confidence measures under the exchangeability assumption that is usually weaker than those traditionally used.

Consider, for example, a set of moving objects (vessels, vehicles, planes, etc.) $z_1, z_2, ...$ and this movement might be normal (typical, conformal) or anomalous (atypical, non-conformal). We make an idealised assumption that $z_1, z_2, ...$ are from the same probability distribution P on the measurable feature space X independent from each other; however no further assumptions are made about P, which is completely unknown.

In this paper, the problem of anomaly detection in the maritime domain deals with trajectories of the ships to detect suspicious behaviours: a sudden change of direction, or speed, or anchoring, etc.

There has been previous research in applying conformal prediction to anomaly detection in the maritime surveillance domain [5]. Those methods focus on non-

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 271–280, 2014.

conformity measures using nearest neighbours with Hausdorff distance or local densities of neighbourhoods with Local Outlier Factor [4,7].

In this paper we experiment with two different measures of non-conformity. In particular, the nearest neighbours non-conformity measure and the kernel density non-conformity measure have been used to detect anomalies. The data was obtained from Automatic Identification System (AIS) – a tracking system for vessels that is used to broadcast the location (retrieved by a GPS receiver onboard) of a vessel over radio-waves every few seconds.

The remaining part of this paper describes some details of conformal predictors including non-conformity measures, efficiency (performance) criteria, then a dimensionality reduction package T-SNE, the description of the data and the results with discussion. In particular, we propose average p-value as a level-independent criterion for assessing the efficiency.

2 Method

2.1 Conformal Prediction

Conformal prediction is a framework that allows making predictions with valid measures of confidence. Conformal Anomaly Detection (CAD) is an extension of Conformal Prediction that focuses on one-class (normal) in the unsupervised or semi-supervised setting [4].

Conformal Anomaly Detection

Input : Non-Conformity Measure A, significance level ϵ, training objects $z_1, z_2, ..., z_{n-1}$ and new object z_n
Output: P-value p_n, boolean variable Anomaly

$D = \{z_1, ..., z_n\}$
for $i \leftarrow 1$ **to** n **do**
  $\alpha_i \leftarrow A(D \setminus z_i, z_i)$
$\tau \leftarrow U(0,1)$
$p_n \leftarrow \frac{|\{i:a_i>a_n\}|+\tau|\{i:a_i=a_n\}|}{n}$
if $p_n < \epsilon$ **then**
  $Anomaly_j \leftarrow$ **true**
else
  $Anomaly_j \leftarrow$ **false**

Basically, the method tests whether a *new object* z_n might be generated by the same distribution as the previous *(training)* objects $z_1, \ldots, z_{n-1}$. If produced *p-value* p_n is small, then the hypothesis of the new object's normalcy is likely to be rejected, so the abnormality is confirmed.

The *significance* level ϵ regulates the pre-determined level of confidence. According to the validity property[2], if all the data objects $z_1, \ldots, z_n$ are really generated by the same distribution, then probability that $p_n \leq \epsilon$ is at most ϵ. In the context of anomaly detection this means that if z_n is not an anomaly, it

will be classified as anomaly with probability at most ϵ. This allows the false positive rate to be calibrated with a significance level parameter ϵ [7].

Another goal is efficiency: if z_n is an *anomaly*, we wish this to be captured by our test assigning a small p-value.

This performance depends on the selection of a *Non-Conformity measure (NCM)* denoted as A – that is a sort of information distance between an object and a set of the same type objects.

In this paper we use leave-one-out cross-validation to evaluate the performance. For each object a p-value is calculated using the rest of the objects as a training set. One advantage of using leave-one-out is independence on the order of data objects. Another is that it allows doing fair cross-validation using dimensionality reduction just once.

2.2 Performance Criterion

The validation of leave-one-out is done with *supervised anomaly detection* which has labelled anomalies and normal objects (from a *testing set*) where the correctness of output can be checked.

As mentioned above, the output (and the performance measure) typically depend on the significance level ϵ. Using a fixed ϵ, the more objects are classified as anomalies, the more sensitive the p-values as a test for randomness.

For supervised anomaly detection, to get an overall performance measure, independent of ϵ we adopt the well-known measure *receiver operating curve* (ROC) and use the– *area under ROC curve* (AUC) . For each value of ϵ we can produce two statistics: the percentage of normal objects classified as normal objects (that is close to $1 - \epsilon$ by validity), and the percentage of captured anomalies. AUC is the area under the corresponding ϵ-parametrized two-dimensional curve inside the square $[0, 1]^2$.

In [7], partial AUC (pAUC) is suggested for conformal anomaly detection, because it is important to minimise the number of false positives. pAUC only considers a subsection of the AUC, in particular false positive rate $\in [0, 0.01]$ and pAUC is normalised such that its outputs are $\in [0, 1]$.

Another important goal is to make the size of the *prediction set* as small as possible. The prediction set is the set of all the possible objects z_n from the feature space such that $p(z_n) \geq \epsilon$. Such kind of performance measure was investigated in the context of anomaly detection by Lei et al. [8].

We propose a new ϵ-independent version of this performance measure called *average p-value* (APV). Average p-value is the p-value of a potential new object, averaged over its location in the feature space. Its approximation can be calculated by using a finite grid of points uniformly spaced out. Every object in the grid will have a p-value calculated using a training set. The training set is fixed for each element of the grid. In the online setting it is possible to generate an APV after each iteration (using the set of normal examples as the training set). In this paper we choose to use all normal and abnormal objects in the training set to match the leave-one-out setting. We recommend using the min and max points from observed data to be used as the corners of the grid. A grid of g^d cells

is generated where g is the grid resolution, and d is number of dimensions of the feature space. A p-value is generated for each cell using the center point of each cell as the object to be evaluated. In this paper we use $g = 100$ and $d = 2$ to give a grid of 100x100 cells.

An alternative setting is the *unsupervised anomaly detection* setting which is designed for when either the data is unlabelled or no examples of anomalous objects have been provided. It considers the whole feature space as an 'ideal' testing set and considers its training set $z_1, \ldots, z_{n-1}$ as normal. In this setting AUC, and pAUC are not applicable as they both require labels, however APV could be used as a criterion in this setting. In the supervised setting AUC is preferable to APV for measuring performance, but APV can still provide information on the efficiency of non-conformity measures.

2.3 Non-conformity Measures

In this work we consider two non-conformity measures: the first is based on Kernel Density Estimation (KDE) and another, for comparison, on Nearest Neighbours algorithm. Lei et al. [8] considered KDE as a conformity measure in the unsupervised setting.

We shall start with the Kernel Density Estimation (KDE) measure. It allows assessing non-conformity based on the density of data points. The normal objects usually are concentrated in a relatively small areas (high density areas or clusters) while anomalies will be outside of these clusters. This can be exploited by estimating a probability density function from empirical data set. A standard method to do this is to use kernel density estimation. It is a non-parametric technique that requires no knowledge of the underlying distribution.

We can interpret a density function as a measure of conformity – many similar type of data points will be located together; hence we can multiply it by minus one to convert it to a non-conformity measure for consistency.

Input : Object z_i, Set of objects $z_1, z_2, ...z_n$ (note in this setup z_i is included in the set), bandwidth h, Kernel function K, number of dimensions d

Output: Non-conformity score A

$$A_i = -\left(\frac{1}{nh^d}\sum_{j=1}^{n} K\left(\frac{z_i - z_j}{h}\right)\right)$$

Kernel density estimators use the previous objects with a bandwidth parameter h that specifies the width of each object.

We will treat the bandwidth uniformly in each dimension, and fixed for each object. A kernel K is a symmetric function centred around each data point. In this work we use a Gaussian Kernel function:

$$K(u) = (2\pi)^{-d/2} e^{-\frac{1}{2}u^T u}$$

Lei et al. [8] have carried out work extending conformal prediction to produce minimal prediction regions with the use of kernel density estimators and

initially proposed KDE as a conformity measure in the unsupervised setting. Their method is underpinned by utilizing a custom bandwidth estimator that minimises the Lebesgue measure of the prediction set in the space.

We have not applied any bandwidth estimators in this paper because we wish to compare KDE with another method that also has a parameter and test performance for the parameters against multiple performance criterion.

We also apply k-Nearest Neighbour (kNN) NCM [5]: d_{ij}^{+} is the jth nearest distance to an object z_i from other objects.

Input : Object z_i, Set of objects $z_1, z_2, ...z_n$, number of nearest neighbours k

Output: Non-Conformity score A

$A_i = \sum_{j=1}^{k} d_{ij}^{+}$

The nearest neighbour non-conformity measure was found to be useful in detecting anomalies [5] and we shall use it to compare performance with the KDE NCM.

2.4 Dimensionality Reduction

The dimensionality of trajectory data is high ($4N$, where N is a number of points in a trajectory) and in order to apply kernel density estimation, we need to decrease the dimensionality of our data.

This is achieved by applying a package called T-SNE – a dimensionality reduction system. The t-Distributed Stochastic Neighbour Embedding (T-SNE) algorithm [9] is a non-deterministic and effective dimensionality reduction algorithm. It has been primarily used for visualisation but we use it to transform our data to lower-dimensional space to evaluate non-conformity measures.

In this particular application of T-SNE to trajectory data we replaced the Euclidean pairwise distance matrix with the Hausdorff distance matrix [4], but otherwise use the standard MATLAB implementation[1]. To remind the reader that the directional Hausdorff distance $\overrightarrow{H}(F,G)$ is the distance from set F to set G. $H(F,G)$ is the symmetrical Hausdorff distance. Hausdorff uses a distance metric *dist* between the sets of points:

$$\overrightarrow{H}(F,G) = \max_{a \in F} \left\{ \min_{b \in G} \{dist(a,b)\} \right\}$$

$$H(F,G) = \max \left\{ \overrightarrow{H}(F,G), \overrightarrow{H}(G,F) \right\}$$

3 Data

An object in our task is a trajectory that can be represented as a function of position over time. We convert the trajectories into a sequence of discrete $4D$ points (x, y, x_{speed}, y_{speed}) in a similar method to [4].

[1] `http://homepage.tudelft.nl/19j49/t-SNE.html`

The original broadcasts are interpolated at a sampling distance of 200m.

If a vessel leaves the observation area for a period of 10 minutes or more, or if the vessel is stationary for a period of 5 minutes or more we consider this as the end of a trajectory. Therefore a *trajectory* is a sequence of $4D$ Points and can have any length. The $4D$ points are normalised such that $x, y \in [0, 1]$ and $x_{speed}, y_{speed} \in [-1, 1]$.

The Portsmouth dataset we evaluate was collected from a single AIS receiver on the south coast of England, during July of 2012 for one week. We filtered the data such that it only contains AIS broadcasts that report their location in a specific area between the Isle of Wight and Portsmouth. This was done to ensure consistency as the further an AIS broadcast travels the more likely it is to be lost and the data becomes less reliable.

In this dataset we consider only passenger, tanker and cargo vessels to reflect a degree of 'regular' behaviour (going from A to B and back). We assume that this data does not contain anomalous behaviour. To add anomalies we artificially inserted two sources of anomalies data points. The first contains 22 search & rescue helicopter trajectories. The other source is 180 'artificial' anomalies: random walks that have been generated starting from a random position of a random observed normal vessel. They follow a random direction and speed and a new point is generated every 200m as it has been suggested in [5]. However, unlike in [4] we only consider the entire trajectory and do not calculate detection delay.

Instead of generating anomalous trajectories of 3km in length we are using different length of "artificial" anomalies. The composition of our 180 'artificial' trajectories is the following: 150 200m long, 20 400m long, 10 600m long, 10 800m long and 10 that are 1000m long. The aim is to diversify the difficulty by providing both easy and difficult anomalies to detect.

The dataset consists of 1124 normal trajectories with 202 anomalies added to it. All these trajectories can be seen in Fig 1.

Prior to applying conformal prediction we run the T-SNE algorithm to produce $2D$ representations of the trajectories.

4 Results

For measuring the performance of the non-conformity measures we use AUC as introduced in section 2.2. The partial AUC (pAUC) is also used to show performance for $fpr \in [0, 0.01]$, note that pAUC is normalised to be in the range [0,1]. The average p-value (APV) introduced in section 2.2 is calculated, recall the lower the APV the more efficient the classifier.

AUC and pAUC are our criteria for anomaly detection ability in the supervised setting and the average p-value in the unsupervised setting which doubles as a measure of efficiency. We compare both non-conformity measures for the best parameter values of AUC, pAUC and APV. The APV, AUC and pAUC for various parameter values of both NCMs can be found in the Table 1.

Table 2 was created to expand upon the k neighbours parameter as it is apparent that the highest AUC k-NN classifier was not in the initial parameter

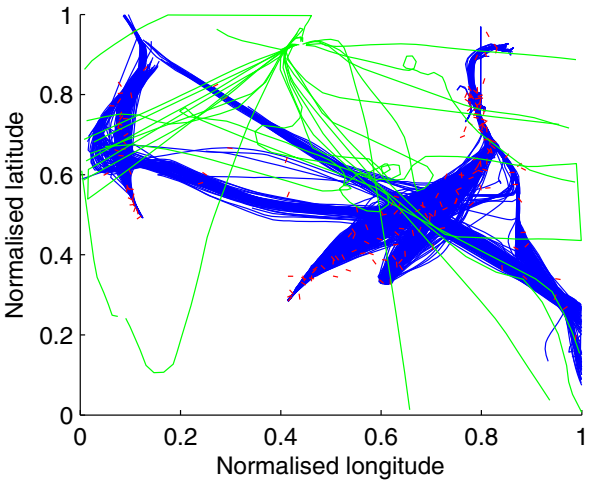


Fig. 1. Blue shows normal trajectories. Red shows the last 200m of the artificial anomalous trajectories. The green trajectories are the helicopters.

set. A rather important thing to note with testing leave-one-out is that anomalies are part of the training set, in practical applications ideally the training set would not contain anomalies. From the tables for all the parameters the highest AUC (supervised setting) are 0.7830 for KDE ($h = 3$) and 0.7616 for kNN ($k = 80$), it is clear that KDE has the higher AUC, and is therefore better at detecting anomalies across all ϵ in the leave-one-out setting. For both these parameters k-NN ($k = 80$) also has a larger APV 0.0638 against KDE ($h = 3$) 0.0606 which indicates that KDE is more efficient and offers better performance than k-NN when AUC is the criterion.

When we consider the most efficient APV (unsupervised setting) as a criterion k-NN's best parameter is $k = 7$ with APV of 0.0453 , whilst KDE's smallest APV is 0.0441 for $h = 1$.

The optimal result for the supervised problem requires more neighbours (k=80) than the unsupervised one (k=7) because most of the anomalies are close to each other (concentrating in a small area on Fig.2) which makes this problem harder. At the same time their influence on the unsupervised prediction is relatively small.

The pAUC Criterion in our leave-one-out setting may not be appropriate as the number of anomalies is far greater than a 1% composition of the dataset, but it is still a vital criterion for the purpose of minimising the false positive rate. KDE's best parameter by pAUC is $h = 2$ with a pAUC 0.484 and k-NN's best pAUC is with $k = 10$ with 0.484, however with these parameters $k = 10$ has a smaller APV and is thus more efficient. k-NN also achieves higher pAUC for more parameter values than KDE. This is quite apparent through with pAUC > 0.03 for $k = 7$ to $k = 20$, and for $k = 40$ to $k = 100$, where as for KDE only $h = \{2, 7, 8\}$ has pAUC above 0.03.

Table 1. AUC, APV and pAUC for various parameters of k-NN and KDE NCMs

k (k-NN) or h (KDE)	1	2	3	4	5	6	7	8	9	10
KDE AUC	0.6116	0.7620	**0.7830**	0.7455	0.6727	0.5932	0.5086	0.4406	0.3811	0.3518
k-NN AUC	0.2977	0.3051	0.3407	0.3611	0.3894	0.4066	0.4193	0.4323	0.4466	0.4618
KDE APV	**0.0441**	0.0519	0.0606	0.0694	0.0801	0.0936	0.1103	0.1307	0.1575	0.1941
k-NN APV	0.0490	0.0481	0.0469	0.0461	0.0458	0.0454	**0.0453**	0.0453	0.0455	0.0456
KDE pAUC	0.0082	**0.0484**	0.0285	0.0235	0.0270	0.0297	0.0415	0.0342	0.0250	0.0000
k-NN pAUC	0.0001	0.0099	0.0099	0.0099	0.0150	0.0276	0.0340	0.0381	0.0427	**0.0484**

Table 2. Extension of k-NN results

k (k-NN)	20	30	40	50	60	70	80	90	100	110
K-NN AUC	0.6157	0.7051	0.7257	0.7403	0.7519	0.7547	**0.7616**	0.6832	0.6301	0.6032
k-NN APV	0.0486	0.0519	0.0549	0.0574	0.0595	0.0615	0.0638	0.0705	0.0827	0.0954
k-NN pAUC	0.0304	0.0253	0.0327	0.0308	0.0400	0.0381	0.0384	0.0434	0.0375	0.0110

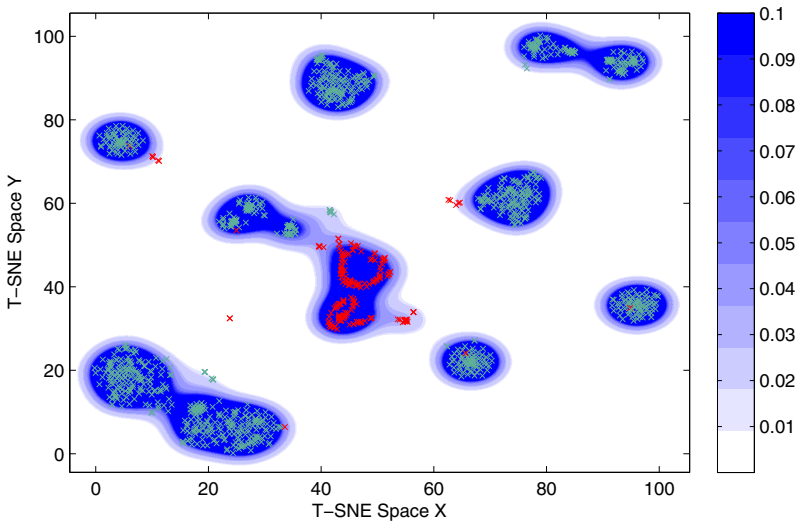


Fig. 2. Prediction sets for various parameters of ϵ in T-SNE space for KDE NCM (h=3). The labelled colours are for various values of ϵ, the red objects are anomalies, and the teal colour is used for the normal trajectories.

Figure 2 visualises the 'normal' class prediction sets of various ϵ for the KDE NCM. It is generated using a grid as the test set and all the objects from our dataset as the training set for each object in the grid.

5 Conclusions and Future Work

In this paper we applied conformal anomaly detection and studied applicable performance measures of efficiency. We have seen that it may be problematic to evaluate the performance of conformal anomaly detection directly because usually either the amount of labelled data for testing the accuracy is small or these data are not representative enough. Therefore we propose average p-value. Average p-value is an as a performance criterion that works in both the supervised and unsupervised settings and does not require labelled anomalies to evaluate performance. At the same time, it is independent on the significance level.

However, we applied some supervised criteria as well.

As examples of NCMs, we used two methods based on the idea of density approximation. One of them is nearest neighbours (k-NN) algorithm and the other is kernel density estimation (KDE) that considers an entire trajectory to the maritime surveillance domain. In addition, we reduced the dimensionality of our dataset to compare the different non-conformity measures.

In the leave-one-out supervised setting KDE NCM for our dataset in the supervised leave-one-out setting has higher AUC than the k-NN NCM. However for most anomaly detection applications performance at small false positive rates is more important. If small false positive rate (in the form of pAUC) is the primary criterion then k-NN NCM performs better than the KDE NCM.

Going to average p-value we see that KDE can lead to more efficient predictions with a smaller average p-value than k-NN, this indicates KDE NCM in the unsupervised setting with a good choice of parameter performs better with our dataset than the k-NN NCM.

Both KDE NCM and k-NN NCM performances for all criterion are dependent on the choice of parameter h and k respectively. We included some observations related to that.

For future work, it would be interesting to continue the work using other sources of data and to reach some explanation of the noticed effects. We also plan to apply various other NCMs in search of anomalous objects.

Acknowledgements. James Smith is very grateful for a PhD studentship jointly funded by Thales UK and Royal Holloway, University of London. This work is supported by EPSRC grant EP/K033344/1 ("Mining the Network Behaviour of Bots"); by the National Natural Science Foundation of China (No.61128003) grant; and by grant 'Development of New Venn Prediction Methods for Osteoporosis Risk Assessment' from the Cyprus Research Promotion Foundation. We are also grateful to Rikard Laxhammar, Vladimir Vovk, Christopher Watkins and Jiaxin Kou for useful discussions. AIS Data was provided by Thales UK.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection A Survey. ACM Computing Surveys (CSUR) (2009), `dl.acm.org`
2. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer (2005)

3. Gammerman, A., Vovk, V.: Hedging predictions in machine learning. The Computer Journal 50(2), 151–163
4. Laxhammar, R., Falkman, G.: Sequential Conformal Anomaly Detection in Trajectories based on Hausdorff Distance. In: 2011 Proceedings of the 14th International Conference on Information Fusion (FUSION) (2011)
5. Laxhammar, R., Falkman, G.: 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, pp. 47–55. ACM (2010)
6. Laxhammar, R., Falkman, G.: Online Detection of Anomalous Sub-trajectories: A Sliding Window Approach Based on Conformal Anomaly Detection and Local Outlier Factor. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 192–202. Springer, Heidelberg (2012)
7. Laxhammar, R.: Conformal anomaly detection: Detecting abnormal trajectories in surveillance applications. PhD Thesis, University of Skovde (2014)
8. Lei, J., Robins, J., Wasserman, L.: Distribution-Free Prediction Sets. Journal of the American Statistical Association 108(501), 278–287 (2013)
9. Van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research 9(11) (2008),
`http://homepage.tudelft.nl/19j49/t-SNE.html`

Rule Extraction with Guaranteed Fidelity

Ulf Johansson[1,⋆], Rikard König[1], Henrik Linusson[1], Tuve Löfström[1], and Henrik Boström[2]

[1] School of Business and IT
University of Borås, Sweden
{ulf.johansson,rikard.konig,henrik.linusson, tuve.lofstrom}@hb.se
[2] Department of Systems and Computer Sciences
Stockholm University, Sweden
henrik.bostrom@dsv.su.se

Abstract. This paper extends the conformal prediction framework to rule extraction, making it possible to extract interpretable models from opaque models in a setting where either the infidelity or the error rate is bounded by a predefined significance level. Experimental results on 27 publicly available data sets show that all three setups evaluated produced valid and rather efficient conformal predictors. The implication is that augmenting rule extraction with conformal prediction allows extraction of models where test set errors or test sets infidelities are guaranteed to be lower than a chosen acceptable level. Clearly this is beneficial for both typical rule extraction scenarios, i.e., either when the purpose is to explain an existing opaque model, or when it is to build a predictive model that must be interpretable.

Keywords: Rule extraction, Conformal prediction, Decision trees.

1 Introduction

When predictive models must be interpretable, most data miners will use decision trees like C4.5/C5.0 [1]. Unfortunately, decision trees are much weaker in terms of predictive performance than opaque models like support vector machines, neural networks and ensembles. Opaque predictive models, on the other hand, make it impossible to assess the model, or even to understand the reasoning behind individual predictions. This dilemma is often referred to as the *accuracy vs. comprehensibility trade-off*.

One way of reducing this trade-off is to apply *rule extraction*, which is the process of generating a transparent model based on a corresponding opaque predictive model. Naturally, extracted models must be as good approximations as possible of the opaque models. This criterion, called *fidelity*, is therefore a

⋆ This work was supported by the Swedish Foundation for Strategic Research through the project High-Performance Data Mining for Drug Effect Detection (IIS11-0053) and the Knowledge Foundation through the project Big Data Analytics by Online Ensemble Learning (20120192).

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 281–290, 2014.

key part of the optimization function in most rule extracting algorithms. For classification, the *infidelity rate* is the proportion of test instances where the extracted model outputs a different label than the opaque model. Similarly, the fidelity is the proportion of test instances where the two models agree. Unfortunately, when *black-box* rule extraction is used, i.e., when the rule extractor utilizes input-output patterns consisting of the original input vector and the corresponding prediction from the opaque model to learn the relationship represented by the opaque model, the result is often a too specific or too general model resulting in low fidelity on the test set, that is, the extracted model is actually a poor approximation of the opaque. Consequently, decision makers would like to have some guarantee *before* applying the extracted model to the test instances that the predictions will actually mimic the opaque.

In conformal prediction [2], prediction sets with a bounded error are produced, i.e., for classification, the probability of excluding the correct class label is guaranteed to be less than the predetermined significance level. The prediction sets can contain one, multiple or even zero class labels, so the price paid for the guaranteed error rate is that not all predictions are informative. In, inductive conformal prediction (ICP) [2], just one model is induced from the training data, and then used for predicting all test instances, but a separate data set (called the *calibration set*) must be used for calculating *conformity scores*.

The conformal prediction framework has been applied to several popular learning schemes, such as ANNs [3], kNN [4] and SVMs [5]. Until now, however, the guarantee provided by conformal prediction has always been related to the error rate. In this paper, we extend the conformal prediction framework to rule extraction, specifically introducing the possibility to bound the infidelity rate by a preset significance level.

2 Background

Rule extraction has been heavily investigated for ANNs, and the techniques have been applied mainly to ANN models; for an introduction and a good survey of traditional methods, see [6]. For ANN rule extraction, there are two fundamentally different extraction strategies, *decompositional* (*open-box* or *white-box*) and *pedagogical* (*black-box*). Decompositional approaches focus on extracting rules at the level of individual units within a trained ANN. Typically, the output of each hidden and output unit is first modeled as a consequent of their inputs, before the rules extracted at the individual unit level are aggregated to form the composite rule set for the ANN. Two classic open-box algorithms are RX [7] and Subset [8].

The core pedagogical idea is to view rule extraction as a learning task, where the target concept is the function originally learned by the opaque model. Black-box rule extraction is therefore an instance of predictive modeling, where each input-output pattern consists of the original input vector $\boldsymbol{x}_i$ and the corresponding prediction $f(\boldsymbol{x}_i; \theta)$ from the opaque model. One typical and well-known black-box algorithm is TREPAN [9].

It must be noted that black-box rule extraction algorithms can be applied to any opaque model, including ensembles, and it can use any learning algorithm producing interpretable models as the actual rule extractor. An inherent problem for open-box methods, regarding both running time and comprehensibility, is the scalability. The potential size of a rule for a unit with n inputs each having k possible values is k^n, meaning that a straightforward search for possible rules is normally impossible for larger networks. Consequently, most modern rule extraction algorithms are black-box, see the more recent survey [10].

There is, however, one very important problem associated with black-box rule extraction. Even if the algorithm aims for maximizing fidelity in the learning phase, there is no guarantee that the extracted model will actually be faithful to the opaque model when applied to test set instances. Instead, since black-box rule extraction is just a special case of predictive modeling, the extracted models may very well overfit or underfit the training data, leading to poor fidelity on test data. The potentially low test set fidelity for black-box techniques stands in sharp contrast to open-box methods where the rules, at least in theory, should have perfect fidelity, even on the test set. Consequently, in situations where a very high fidelity is needed, open-box methods may be necessary; see e.g., [11]. Ideally though, we would like to have the best of both worlds, i.e., providing the efficiency and the freedom to use any type of opaque model present in black-box rule extractors, while guaranteeing test set fidelity. Again, the purpose of this paper is to show how the conformal prediction framework can be employed for achieving this.

An interesting discussion about the purpose of rule extraction is found in [12], where Zhou argues that rule extraction really should be seen as two very different tasks; rule extraction *for* neural networks and rule extraction *using* neural networks[1]. While the first task is solely aimed at understanding the inner workings of an opaque model, the second task is explicitly aimed at extracting a comprehensible model with higher accuracy than a comprehensible model created directly from the data set. More specifically, in rule extraction *for* opaque models, the purpose is most often to explain the reasoning behind individual predictions from an opaque model, i.e., the actual predictions are still made by the opaque model. In that situation, test set fidelity must be regarded as the most important criterion, since we use the extracted model to understand the opaque. In rule extraction *using* opaque models, the predictions are made by the extracted model, so it is used both as the predictive model and as a tool for understanding and analysis of the underlying relationship. In that situation, predictive performance is what matters, so the data miner must have reasons to believe that the extracted model will be more accurate than other comprehensible models induced directly from the data. The motivation for that rule extraction *using* opaque models may work is that even a highly accurate opaque model is a smoothed representation of the underlying relationship. In fact, train-

[1] Naturally this distinction is as relevant for rule extraction from any opaque model, not just from ANNs, so we use the terms rule extraction for or using *opaque models* instead.

ing instances misclassified by the opaque model are often atypical, i.e., learning such instances will reduce the generalization capability. Consequently, rule extraction is most often less prone to overfitting than standard induction, resulting in smaller and more general models.

2.1 Conformal Prediction

A key component in ICP is the conformity function, which produces a score for each instance-label pair. When classifying a test instance, scores are calculated for all possible class labels, and these scores are compared to scores obtained from a calibration set consisting of instances with known labels. Each class is assigned a probability that it does conform to the calibration set based on the fraction of calibration instances with a higher conformity score. For each test instance, the conformal predictor outputs a set of predictions with all class labels having a probability higher than some predetermined *significance level*. This prediction set may contain one, several, or even no class labels. Under very general assumptions, it can be guaranteed that the probability of excluding the true class label is bounded by the chosen significance level, independently of the conformity function used, for more details see [2].

In ICP, the conformity function A is normally defined relative to a trained model M:

$$A(\langle \bar{x}, c \rangle) = F(c, M(\bar{x})) \tag{1}$$

where $\bar{x}$ is a vector of feature values (representing the example to be classified), c is a class label, $M(\bar{x})$ returns the class probability distribution predicted by the model, and the function F returns a score calculated from the chosen class label and predicted class distribution.

Using a conformity function, a *p-value* for an example $\bar{x}$ and a class label c is calculated in the following way:

$$p_{\langle \bar{x}, c \rangle} = \frac{|\{s : s \in S \wedge A(s) \leq A(\langle \bar{x}, c \rangle)\}|}{|S|} \tag{2}$$

where S is the calibration set. The prediction for an example $\bar{x}$, where $\{c_1, \ldots, c_n\}$ are the possible class labels, is:

$$P(\bar{x}, \sigma) = \{c : c \in \{c_1, \ldots, c_n\} \wedge p_{\langle \bar{x}, c \rangle} > \sigma\} \tag{3}$$

where σ is a chosen significance level, e.g., 0.05.

3 Method

The purpose of this study is to extend the conformal prediction framework to rule extraction, and show how it can be used for both rule extraction *for* opaque models and rule extraction *using* opaque models. Since standard ICP is used, the difference between the scenarios is just how the calibration set is used. For the final modeling, all setups use J48 trees from the Weka workbench [13]. Here J48,

which is the Weka implementation of C4.5, uses default settings, but pruning was turned off and Laplace smoothing was used for calculating the probability estimates. The three different setups evaluated are described below:

- **J48:** J48 trees built directly from the data. When used as a conformal predictor, the calibration set uses the true targets, i.e., the guarantee is that the error rate is bounded by the significance level.
- **RE-a:** Rule extraction *using* opaque models. Here, an opaque model is first trained, and then a J48 tree is built using original training data inputs, but with the predictions from the opaque model as targets. For the conformal prediction, the calibration set uses the true targets, so the guarantee is again that the error rate is bounded by the significance level.
- **RE-f:** Rule extraction *for* opaque models. The J48 model is trained identically to RE-a, but now the conformal predictor uses predictions from the opaque model as targets for the calibration. Consequently, the guarantee is that the infidelity rate will be lower than the significance level.

In the experimentation, bagged ensembles of 15 RBF networks were used as opaque models. With guaranteed validity, the most important criterion for comparing conformal predictors is *efficiency*. Since high efficiency roughly corresponds to a large number of singleton predictions, *OneC*, i.e., the proportion of predictions that include just one single class, is a natural choice. Similarly, *MultiC* and *ZeroC* are the proportions of predictions consisting of more than one class, and empty predictions, respectively. One way of aggregating these number is *AvgC*, which is the average number of classes in the predictions.

In this study, the well-known concept of *margin* was used as the conformity function. For an instance i with the true class Y, the higher the probability estimate for class Y the more conforming the instance, and the higher the other estimates the less conforming the instance. For the evaluation, 4-fold cross-validation is used. The training data was split 2:1; i.e., 50% of the available instances were used for training and 25% were used for calibration. The 27 data sets used are all publicly available from either the UCI repository [14] or the PROMISE Software Engineering Repository [15].

4 Results

Table 1 below shows the accuracy, AUC and size (total number of nodes) for the J48 models produced using either standard induction or rule extraction. As described in the introduction, the rule extraction is supposed to increase model accuracy, produce smaller models, or both. Comparing mean values and wins/ties/losses, the results show that the use of rule extraction actually produced models with higher accuracy. A standard sign test requires 19 wins for significance when $\alpha = 0.05$, so the difference is statistically significant at that level. Looking at models sizes, the extracted models are also significantly less complex. When comparing the ranking ability, however, the larger induced tree models obtained higher AUCs, on a majority of the data sets.

Table 1. Accuracy, AUC and Size

	Accuracy		AUC		Size			Accuracy		AUC		Size	
Data set	Ind.	Ext.	Ind.	Ext.	Ind.	Ext.	Data set	Ind.	Ext.	Ind.	Ext.	Ind.	Ext.
ar1	.909	.913	.457	.608	8.0	6.3	kc1	.839	.848	.680	.595	100.0	10.8
ar4	.817	.808	.664	.660	8.5	5.8	kc2	.827	.828	.785	.674	26.5	8.0
breast-w	.921	.928	.953	.945	20.8	15.0	kc3	.889	.903	.688	.629	25.5	4.8
colic	.705	.717	.713	.731	34.8	21.8	letter	.824	.800	.838	.824	20.0	23.3
credit-a	.712	.751	.771	.800	57.5	32.5	liver	.578	.620	.561	.610	22.3	23.5
credit-g	.683	.712	.620	.643	108.0	51.3	mw1	.901	.917	.679	.616	15.5	4.5
cylinder	.644	.634	.630	.638	63.3	50.3	sonar	.618	.680	.657	.733	18.8	14.0
diabetes	.691	.711	.690	.684	33.3	29.5	spect	.771	.793	.699	.731	25.0	11.8
heart-c	.719	.723	.754	.773	31.0	21.8	spectf	.744	.756	.718	.691	20.3	13.0
heart-h	.760	.786	.769	.804	28.5	14.5	tic-tac-toe	.770	.694	.775	.631	52.8	45.5
heart-s	.767	.748	.810	.784	31.3	17.3	vote	.899	.905	.933	.928	19.8	13.5
hepatitis	.781	.781	.746	.701	18.5	14.0	vowel	.786	.725	.804	.782	13.5	15.3
iono	.769	.789	.732	.750	13.3	13.8	**Mean**	**.761**	**.767**	**.719**	**.712**	**31.9**	**19.0**
jEdit4042	.642	.639	.669	.671	21.3	14.8	**Wins**	**8**	**19**	**15**	**12**	**4**	**23**
jEdit4243	.583	.589	.606	.599	23.5	15.8							

Turning to the results for conformal prediction, Fig 1 shows the behavior of extracted J48 trees as conformal predictors on the Iono data set, when using true targets for calibration. Since the conformal predictor is calibrated using true targets, it is the error and not the infidelity that is bounded by the significance level.

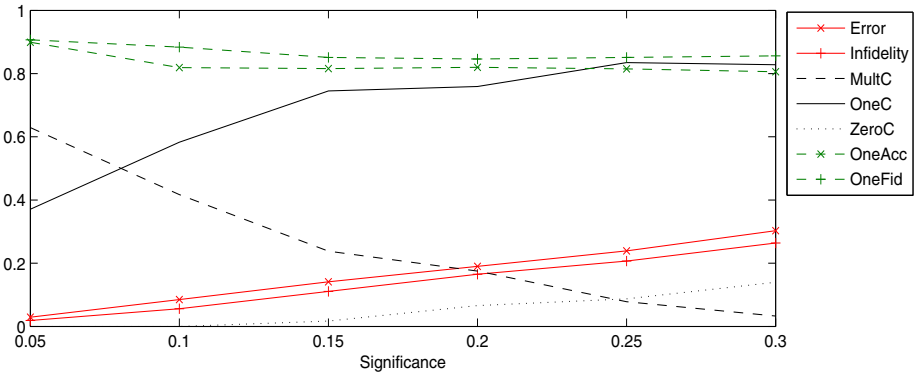


Fig. 1. Rule extraction *using* opaque model. Iono.

First of all, the conformal predictor is valid and well-calibrated, since the error rate is very close to the corresponding significance level. Analyzing the efficiency, the number of singleton predictions (OneC) starts at approximately

40% for $\epsilon = 0.05$, and then rises quickly to over 70% at $\epsilon = 0.15$. The number of multiple predictions (MultiC), i.e., predictions containing both classes, has the exact opposite behavior. The first empty predictions (ZeroC) appear at $\epsilon = 0.10$. Interestingly enough, OneAcc (the accuracy of the singleton predictions) is always higher than the accuracy of the underlying tree model (0.769), so singleton predictions from the conformal predictor could be trusted more than predictions from the original model. Finally, the fidelity of the singleton predictions (OneFid) is very high, always over 80%. In fact, the infidelity rate is always lower than the error, indicating that the extracted conformal predictor is very faithful to the opaque model, even if this is not enforced by the conformal prediction framework in this setup.

Fig 2 below shows the behavior of extracted J48 trees as conformal predictors on the Iono data set, when using the ensemble predictions as targets for calibration.

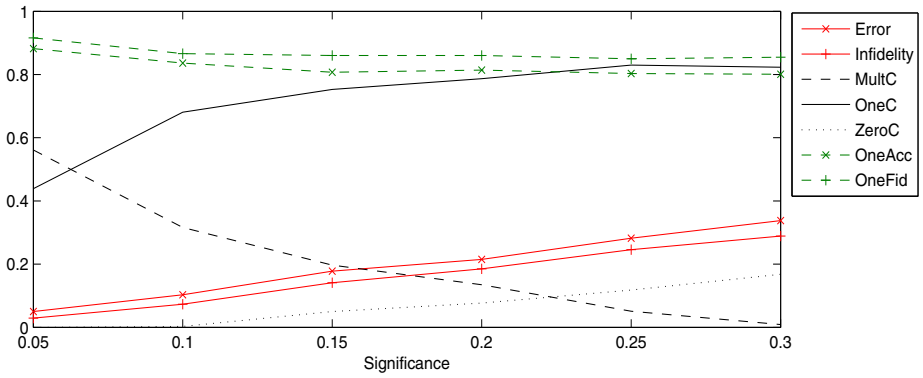


Fig. 2. Rule extraction *for* opaque model. Iono.

In this setup, it is the infidelity and not the error that is guaranteed, and indeed the actual infidelity rate is very close to the significance level. Here, singleton predictions are more common than for the other setup, i.e., it is easier to have high confidence in predictions about ensemble predictions than true targets. The error rate is slightly higher than the significance level, but interestingly enough both OneAcc and OneFid are comparable to the results for the previous setup.

Table 2 below shows detailed results for the three different conformal prediction setups, when the level of significance is $\epsilon = 0.10$. Investigating the errors and infidelities, it is obvious that the conformal prediction framework applies to both rule extraction scenarios, i.e., when the error rate or the infidelity rate must be lower than the significance level. On almost all data sets, the errors for J48 and RE-a are quite close to the significance level $\epsilon = 0.1$, indicating that the conformal predictors are valid and well-calibrated. Similarly, the infidelities for RE-f are also close to 0.1, on most data sets. Looking at the efficiency, measured

Table 2. Conformal prediction with $\epsilon = 0.1$. Bold numbers indicate criteria that are guaranteed by the conformal prediction framework.

	Error			Infidelity		OneC			OneAcc			OneFid	
	J48	RE-a	RE-f	RE-a	RE-f	J48	RE-a	RE-f	J48	RE-a	RE-f	RE-a	RE-f
ar1	**.070**	**.070**	.124	.054	**.091**	.904	.821	.842	.919	.939	.930	.957	.950
ar4	**.043**	**.052**	.083	.042	**.056**	.369	.366	.649	.757	.870	.907	.881	.917
breast-w	**.090**	**.094**	.105	.095	**.102**	.936	.886	.852	.940	.944	.948	.946	.952
colic	**.098**	**.094**	.143	.063	**.086**	.538	.503	.652	.800	.797	.770	.870	.861
credit-a	**.094**	**.111**	.182	.055	**.089**	.556	.644	.818	.803	.818	.770	.908	.885
credit-g	**.104**	**.085**	.195	.024	**.090**	.440	.409	.745	.753	.780	.733	.935	.880
cylinder	**.097**	**.099**	.122	.073	**.093**	.317	.344	.427	.664	.695	.698	.764	.785
diabetes	**.083**	**.096**	.191	.044	**.098**	.415	.447	.723	.795	.774	.734	.888	.865
heart-c	**.073**	**.091**	.127	.045	**.084**	.407	.452	.586	.793	.787	.770	.879	.857
heart-h	**.083**	**.080**	.163	.034	**.075**	.509	.565	.858	.786	.845	.818	.931	.914
heart-s	**.070**	**.072**	.119	.037	**.059**	.461	.458	.625	.852	.834	.804	.915	.897
hepatitis	**.045**	**.055**	.081	.032	**.048**	.528	.445	.578	.925	.853	.851	.936	.914
iono	**.078**	**.089**	.108	.090	**.096**	.570	.608	.625	.845	.850	.805	.806	.813
jEdit4042	**.091**	**.089**	.197	.022	**.062**	.369	.291	.619	.717	.659	.662	.908	.880
jEdit4243	**.083**	**.080**	.265	.015	**.068**	.245	.221	.665	.647	.620	.627	.940	.888
kc1	**.093**	**.097**	.217	.002	**.095**	.784	.729	.900	.881	.866	.870	.997	.997
kc2	**.088**	**.100**	.216	.011	**.101**	.758	.691	.934	.883	.853	.840	.985	.969
kc3	**.075**	**.081**	.157	.011	**.088**	.878	.931	.916	.920	.913	.920	.989	.995
letter	**.098**	**.089**	.098	.082	**.090**	.657	.650	.657	.860	.853	.851	.871	.865
liver	**.094**	**.072**	.170	.042	**.104**	.253	.263	.516	.626	.724	.665	.851	.783
mw1	**.091**	**.091**	.129	.047	**.089**	.963	.990	.932	.911	.919	.935	.967	.983
sonar	**.070**	**.108**	.072	.091	**.089**	.233	.423	.303	.702	.737	.729	.788	.701
spect	**.070**	**.088**	.158	.026	**.056**	.549	.665	.844	.875	.866	.819	.960	.934
spectf	**.078**	**.085**	.114	.066	**.085**	.491	.444	.559	.826	.808	.805	.843	.807
tic-tac-toe	**.105**	**.087**	.235	.023	**.116**	.635	.370	.794	.792	.758	.707	.926	.857
vote	**.096**	**.079**	.091	.062	**.077**	.875	.845	.869	.923	.939	.934	.954	.946
vowel	**.083**	**.064**	.078	.100	**.097**	.581	.458	.378	.836	.854	.805	.777	.724
Mean	**.083**	**.085**	**.146**	**.048**	**.085**	**.564**	**.553**	**.699**	**.816**	**.821**	**.804**	**.903**	**.882**
Mean Rank	-	-	-	-	-	**2.19**	**2.41**	**1.41**	**1.85**	**1.81**	**2.33**	**1.26**	**1.74**

using the OneC metric, RE-f is clearly the most efficient conformal predictor. An interesting observation is that the errors for RE-f often are much higher than the corresponding significance level, thus indicating that the extracted model quite often is certain about the prediction from the ensemble, even when the ensemble prediction turns out to be wrong. This phenomenon is also obvious from the lower OneAcc exhibited by RE-f. Regarding infidelities and OneFid, it may be noted that RE-a turns out to be overly conservative. This actually results in a higher OneFid, compared to RE-f, but the explanation is the much fewer singleton predictions. Simply put, with a high demand on confidence in the selected singleton predictions, these tend to be predicted identically by the ensemble.

Table 3 below shows a summary, presenting averaged values and mean ranks over all data sets for three different significance levels. Included here is the metric AvgC, which is the average number of labels in the prediction sets. Since there are very few empty predictions at $\epsilon = 0.05$, OneC and AvgC will, for this significance level, produce the same ordering of the setups.

Table 3. Conformal prediction summary. Bold numbers indicate criteria that are guaranteed by the conformal prediction framework.

	$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
	Ind	RE-a	RE-f	Ind	RE-a	RE-f	Ind	RE-a	RE-f
Error	**.034**	**.034**	.084	**.083**	**.085**	.146	**.184**	**.183**	.251
Infidelity	-	.018	**.035**	-	.046	**.084**	-	.124	**.190**
AvgC	1.66	1.70	1.46	1.43	1.44	1.26	1.15	1.15	1.01
Rank	2.11	2.70	1.19	2.30	2.48	1.22	2.48	2.33	1.19
OneC	.339	.297	.525	.564	.552	.701	.772	.778	.821
Rank	2.11	2.70	1.19	2.19	2.41	1.41	2.15	2.04	1.81
OneAcc	.772	.752	.778	.815	.819	.805	.794	.796	.794
Rank	1.78	1.89	2.33	1.85	1.89	2.26	2.07	1.93	2.00
OneFid	-	.824	.857	-	.906	.884	-	.878	.869
Rank	-	1.44	1.56	-	1.22	1.78	-	1.48	1.52

Even when analyzing all three significance levels, all conformal predictors seem to be valid and reasonably well-calibrated. Looking for instance at RE-a, the averaged errors over all data sets are 0.034 for $\epsilon = 0.05$, 0.084 for $\epsilon = 0.1$ and 0.183 for $\epsilon = 0.2$. Similarly, the averaged infidelities for RE-f are 0.035 for $\epsilon = 0.05$, 0.084 for $\epsilon = 0.1$ and 0.190 for $\epsilon = 0.2$.

Comparing efficiencies, RE-f is significantly more efficient, with regard to both OneC and AvgC, than the other two setups. J48 and RE-a have comparable efficiencies. Regarding OneAcc, J48 and RE-a are most often more accurate than RE-f. It must, however, be noted that RE-f has a fundamentally different purpose than RE-a and J48, so RE-a should only be compared directly to J48; they are both instances of, in Zhou's terminology, rule extraction *using* opaque models, while RE-f, is rule extraction *for* opaque models. Consequently the most important observation is that all setups have worked as intended, producing valid, well-calibrated and rather efficient conformal predictors for the two different rule extraction scenarios.

5 Concluding Remarks

In this paper, which should be regarded as a proof-of-concept, conformal prediction has been extended to rule extraction *for* opaque models and rule extraction *using* opaque models. The results show that conformal prediction enables extraction of efficient and comprehensible models, where either the error rate or

the infidelity rate is guaranteed. This represents an important addition to the rule extraction tool-box, specifically addressing the problem with a potentially poor test set fidelity present in most black-box rule extractors.

For some reason rule extraction has not been extensively used on regression models, so the next step is to apply conformal prediction to this. We believe that the prediction intervals produced by conformal prediction regression will be a natural part of making extracted regression models accurate and comprehensible.

References

1. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann (1993)
2. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer-Verlag New York, Inc. (2005)
3. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. Tools in Artificial Intelligence 18, 315–330 (2008)
4. Nguyen, K., Luo, Z.: Conformal prediction for indoor localisation with fingerprinting method. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 214–223. Springer, Heidelberg (2012)
5. Makili, L., Vega, J., Dormido-Canto, S., Pastor, I., Murari, A.: Computationally efficient svm multi-class image recognition with confidence measures. Fusion Engineering and Design 86(6), 1213–1216 (2011)
6. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl.-Based Syst. 8(6), 373–389 (1995)
7. Rudy, H.L., Lu, H., Setiono, R., Liu, H.: Neurorule: A connectionist approach to data mining, 478–489 (1995)
8. Fu, L.: Rule learning by searching on adapted nets. In: AAAI, pp. 590–595 (1991)
9. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in Neural Information Processing Systems, pp. 24–30. MIT Press (1996)
10. Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models. FETEW Research Report KBI 0612, K. U. Leuven (2006)
11. Martens, D., Huysmans, J., Setiono, R., Vanthienen, J., Baesens, B.: Rule extraction from support vector machines: An overview of issues and application in credit scoring. In: Rule Extraction from Support Vector Machines, pp. 33–63 (2008)
12. Zhou, Z.H.: Rule extraction: using neural networks or for neural networks? J. Comput. Sci. Technol. 19(2), 249–253 (2004)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
14. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
15. Sayyad Shirabad, J., Menzies, T.: PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada (2005)

Conformal Prediction under Probabilistic Input

Ilia Nouretdinov

Computer Learning Research Centre,
Royal Holloway University of London

Abstract. In this paper we discuss a possible approach to confident prediction from data containing missing values presented in a probabilistic form. To achieve this we revise and generalize the notion of credibility known in the theory of conformal prediction.

1 Introduction

The task of machine learning is to predict a label for a new (or a testing) example x_{l+1} from a given training set of feature vectors $x_1, x_2, \ldots, x_l$ supplied with labels $y_1, y_2, \ldots, y_l$. The conformal prediction technique introduced in [1] and had many applications and extensions later. It allows to make a valid confident prediction.

Originally it was introduced for supervised machine learning problem with clear data structure. But in many practical problems data representation may be complex and combine multiple sorts of information. The conformal prediction was extended in previous works. In [6], it was *semi-supervised learning* when only some examples are presented with labels. In [7] training labels were available only for one of two classes. In [8] an unsupervised learning problem of *anomaly detection* was considered. Another kind of the task is Vapnik's Learning under privileged information [4] that can be interpreted as having missing values in testing examples. A conformal approach to it was made in the work [5].

The direction presented here is probabilistic representation of feature vectors or labels. Assume that there is kind of a priori distribution on features and/or labels. For example it is concentrated at one value when a feature is presented, it is uniform when it is completely missing, and other distributions are applicable when it is known partially or hypothetically. This means neither to try to exclude examples with missing values nor to fill them in a unique way.

An approach to this task is based on the notion of credibility that appears in the standard (supervised) conformal prediction. Unlike the confidence assigned to a likely hypothesis about the new example's label, the credibility answers the question whether any of these hypotheses is true at all. So the credibility is a characteristic of an unfinished data sequence, that includes a new example without its label. This can be naturally extended to the task when some part of training information is missing.

As an area of application needed for an illustration of the proposed method, we take LED data set from UCI repository [2], because a priori distribution on the values has a clear sense for these data.

L. Iliadis et al. (Eds.): AIAI 2014 Workshops, IFIP AICT 437, pp. 291–300, 2014.

2 Machine Learning Background

2.1 Conformal Prediction

Let us remind the properties of conformal prediction (in the case of classification) according to [1].

Assume that each data example z_i consists of x_i that is an m-dimensional vector $x_i = (x_{i1}, \ldots, x_{im})$ and a label y_i that is an element of a finite set Y.

Conformal predictor in supervised case assigns p-value (the value of a test for randomness) to a data sequence

$$p(y) = p((x_1, y_1), \ldots, (x_l, y_l); (x_{l+1}, y))$$

$$= \frac{card\{i = 1, \ldots, l+1 : \alpha_i \geq \alpha_{l+1}\}}{l+1}$$

where $x_1, \ldots, x_l \in X$ are feature vectors of training examples with known classifications $y_1, \ldots, y_l \in Y$, x_{l+1} is a new or testing example with a *hypothetical* label y, and

$$\alpha_i = A(\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, y)\}, (x_i, y_i))$$

for a *nonconformity measure* A that is a strangeness function of a set of labeled feature vectors and one of its elements.

The plan is to check each possible hypothesis about the label of a new example, and to the label of new example would conform the assumption of exchangeability, or with which label the example 'fits well' into the training set? The prediction set consists of satisfactory hypotheses y such that $p(y)$ exceeds a *significance level* γ. The calculations of prediction regions are based on a special function called *nonconformity measure (NCM)* that reflects how strange an example is with respect to others. Then p-value is assigned to each y.

There are two ways to present the results. One of them is the *prediction set*: a list of y which meet this confidence requirement $p(y) \geq \gamma$. The *validity* property implies that the probability of error is at most $1 - \gamma$ whenever the i.i.d. assumption is true. Here an error means true value of y_n being outside the prediction set.

Alternatively we can provide the prediction of a new label together with measures of its individual *confidence*. The correspondence between two types of output is that the confidence is the highest confidence level at which the prediction region consists of (at most) one value. In terms of p-values assigned to different labels, the confidence is a complement to 1 of the second highest p-value.

An individual prediction is also naturally completed with *credibility* that is the first highest p-value. If the credibility is low this means that any existing hypothesis about the label of the new object is unlikely. In other words, the new object itself is not credible enough as a continuation of the data sequence, and this could be said before its label is known. So it can be understood as dealing with an unknown testing label.

Our aim is to extend this idea, dealing other sort of incomplete information in analogous way.

2.2 Standard Credibility

In this work we call credibility a measure of conformity of an incomplete data sequence.

Originally it was applied to the data sequences of the following type:

$$(x_1,y_1),\ldots,(x_l,y_l),x_{l+1}.$$

with y_{l+1} missing.

The credibility is obtained by maximization conformal p-values over all its possible completions:

$$p_{cred}((x_1,y_1),\ldots,(x_l,y_l),x_{l+1})$$

$$= \max_{y_{l+1}\in Y} p((x_1,y_1),\ldots,(x_l,y_l),(x_{l+1},y_{l+1})).$$

The validity property of conformal prediction regions can be easily extended to the credibility. If a data sequence $(x_1,y_1),\ldots,(x_{l+1},y_{l+1})$ is generated by $P = P_1^{l+1}$ where P_1 is a distribution on $X \times \{0,1\}$, then

$$P\{p_{cred}((x_1,y_1),\ldots,(x_l,y_l),x_{l+1}) \le \gamma\} \le \gamma$$

for any $\gamma \in (0,1)$.

In this form it was assumed that the incomplete sequence is obtained from the complete one by forgetting y_{l+1}. In other words it could be said that the incomplete sequence $(x_1,y_1),\ldots,(x_l,y_l),x_{l+1}$ is generated by $P^l \times P_X$ where P_X is the marginal distribution of the feature vector averaged over Y.

2.3 Extensions of Credibility

As we have seen, the standard credibility is the p-value (test for randomness) assigned to an incomplete sequence of examples. Incompleteness means there that the label of the last example is totally missing.

Sometimes a similar aproach can be applied to other kinds of missing values. A close problem is having an unknown feature (not a label) of a new (testing) example. This task is equivalent to learning under privileged (additional) information framework formulated in [4]. The conformal approach of this task was developed in the work [5]. An analogue of credibility was assigned to the sequence

$$(x_1,x_1^*,y_1),\ldots,(x_l,x_l^*,y_l),(x_{l+1},y)$$

with x_{l+1}^* unknown. The feature x^* was called privileged bacause it is available for the training examples.

Next step might be related to missing values in training examples. But a straightforward approach to this task (maximizing p-value over possible fillings of the gap) is not effective because the conformal predictor concentrates on the conformity of the testing example without checking training examples for strangeness. Therefore we would like to consider missing values as distributions.

3 Conformal Approach for Probabilistic Input

For convenience of presentation, in this section we will start form the case of unclear information about binary labels y_i presented in a probabilistic form of a priori distribution. Then we will show how to apply it in a more general case.

3.1 Task and Assumptions

Suppose that $Y = \{0,1\}$, but some information about $y_1,\dots,y_l$ is missing. However, for each $i = 1,\dots,l$ we know that p_i has a meaning of probability that $y_i = 1$. As for y_{l+1}, we assume that it is known as a hypothesis according to the conformal prediction procedure.

How to state this task in a well-defined way and what would be a proper analogue of the i.i.d. assumption in this case?

A mechanism should generate both the 'true' data sequence (including hidden values of y_i) and the 'visible' one (with probabilistic values p_i). This means that the triple (x_i, y_i, p_i) is generated simultaneously. But some agreement between p_i and y_i is also needed so that probabilistic values p_i make sense as probabilities.

To define this formally, assume that P_1 is a distribution on $X \times (0,1)$ and Θ is the uniform distribution on (0,1). First, $P = (P_1 \times \Theta)^{l+1}$ generates

$$(x_1, p_1, \theta_1), \dots, (x_{l+1}, p_{l+1}, \theta_{l+1}).$$

Setting $y_i = 1$ if $p_i < \theta_i$ and $y_i = 0$ otherwise, we can also say that P^* generates a sequence of triples

$$(x_1, p_1, y_1), (x_2, p_2, y_2), \dots, (x_{l+1}, p_{l+1}, y_{l+1})$$

where p_i is 'visible' label and y_i is the 'hidden' one, y_i is stochastically obtained from p_i.

3.2 Special Credibility

In order to make a conformal prediction of y_{l+1} for x_{l+1} we need to consider different hypotheses about it. When a hypothesis is chosen, we work with 'visible' labels $p_1,\dots,p_l$ for training examples and for a 'hidden' value y_{l+1} for the new one. Thus the task is to assign a valid credibility value for a sequence $(x_1, p_1), \dots, (x_l, p_l); (x_{l+1}, y_{l+1})$.

Fix a parameter $s > 0$ called *allowance* which is a trade-off between testing the hypothetical new label with respect to a version of the training data set, and testing the training data set with respect to a priori distribution on missing values.

Suppose that Q is the conditional distribution of $p((x_1,y_1),\dots,(x_l,y_l),(x_{l+1},y_{l+1}))$ given $(p_1,\dots,p_l)$ and y_{l+1}, $q_0 = q_0(p_1,\dots,p_l)$ is the smallest q such that

$$Q\{p > q | p_1,\dots,p_l; y_{l+1}\} \le s$$

and

$$p_{cred} = p_{cred}\left((x_1,p_1),\dots,(x_l,p_l);(x_{l+1},y_{l+1})\right) = q_0(p_1,\dots,p_l;y_{l+1}) + s.$$

Proposition 1. *Assume tgat* $(x_1,p_1),\ldots,(x_l,p_l)$ *and* $(x_{l+1},y_{l+1}$ *are generated by the mechanism described in Section 3.1 and* p_{cred} *us calculated as in Section 3.2, then*

$$P\{p_{cred} \leq \gamma\} \leq \gamma$$

for any $\gamma \in (0,1)$.

Proof: Recall that $p = p(y_1,\ldots,y_l) > q = q(p_1,\ldots,p_l)$ with probability at most s for any given $p_1,\ldots,p_l$. On the other hand, p is valid as a standard conformal predictor's output thus $p \leq \gamma - s$ with overall probability at most $\gamma - s$. Therefore $\gamma - s < p < q$ with probability at least $1 - s - (\gamma - s) = 1 - \gamma$ and probability that $q + s < \gamma$ is bounded by γ. □

3.3 Missing Values in Features

For convenience of presentation we earlier assumed that the labels $y_1,\ldots,y_l$ are given in probabilistic form, although this can be extended to the objects $x_1,\ldots,x_l$ as well.

So let us now assume that P^* generates (H_i,x_i,y_i) where 'visible' H_i is a distribution on X, while 'hidden' $x_i \in X$ is randomly generated by H_i.

If x_i is known clearly, this means that H_i is a distribution concentrated at one point. Otherwise H_i can be understood as an a priori distribution on its missing values. If X is discrete, then H_i can be presented in a vector form.

The extended credibility is defined by analogy. Suppose that Q is the conditional distribution of $p((x_1,y_1),\ldots,(x_l,y_l),(x_{l+1},y_{l+1}))$ given $(H_1,\ldots,H_l,H_{l+1})$ and a fixed y_{l+1}, q_0 is the smallest q such that

$$Q\{p > q | H_1,\ldots,H_{l+1};y_{l+1}\} \leq s$$

and

$$p_{cred}\,((H_1,y_1),\ldots,(H_l,y_l),(H_{l+1},y_{l+1})) = q_0 + s.$$

Obviously an analogue of Proposition 1 is also true in this case.

3.4 Efficient Approximation

To find q_0 exactly one has to know the condition distribution of p given 'visible' data. For the aims of computational efficiency this distribution can be replaced with an empirical one, using Monte-Carlo approximation. Let $H_1 \times H_2 \times \ldots \times H_{l+1}$ generate a large amount of vectors $(x_1,\ldots,x_{l+1})$ and calculate conformal p-value for each of them. Then we will get an empirical distribution of p that allows to estimate q_0 by sorting these p-values and taking one with corresponding rank. An example will be given in Section 4.3.

4 Experiments

For the experiments we use benchmark LED data sets generated by a program from the UCI repository[2].

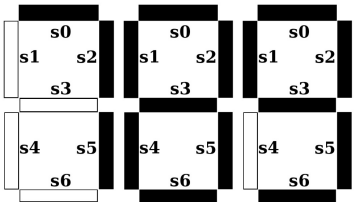


Fig. 1. Canonical images of 7,8,9 in LED data

LED means light emitting diode electronic display. A LED image of a digits has of 7 binary features (pixels). The task is to predict a digit from an image in the seven-segment display. Each of digits $0,1,\ldots,9$ has a canonical image that should normally represent it. Few examples are presented on Fig.1.

Assume now that visible displays can contain mistakes. Each pixel can occasionally show 'on' instead of 'off' or vice versa with probability p_0. For our example we assume that $p_0 = 0.1$ although normally it is much less. The data generating program first randomly selects a canonically represented digit then each of the attributes is inverted with a probability of noise p_0 and the noisy example is added to the data set.

In the work [3] the conformal approach was applied to LED data in its standard supervised form. Now we make some changes in the data statement. First, the probability p_0 itself is known for us. T his means that all values in the training set are probabilistic ones. When we see that a pixel is 'on' this in fact means that it is on with probability $1 - p_0$ and 'off' with p_0, and vice versa. Second, in the testing examples there are no mistakes (as if $p_0 = 0$). The task is to classify a testing example with full information after training on the examples with probabilistic information. It is assumed that the canonical representations are not available for the learner, who has to make predictions based only on the examples with possible mistakes as they are presented in the data.

For experiments we generate some amount of LED digits. The number and distribution (frequency) of labels (0,1,2,...,9) is not restricted, we borrow it from well-known USPS (US Postal Service) benchmark data set in order to have imbalanced classes. Size of the classes is shown in Table 1.

Table 1. Size of different training classes

Class label (digit)	0	1	2	3	4	5	6	7	8	9	Total
Number of examples	359	264	198	166	200	160	170	147	166	177	2007

For a training example, given a label, we take its canonical LED image and make an error in each of the feature with probability γ. In the most of experiments $\gamma = 0.1$ unless stated another.

Testing examples are not probabilistic by the task, so in principle we can make predictions on $2^7 = 128$ possible images. This number includes 10 canonical images of digits.

Later we will consider two types of testing set. A proper one is generated with the same distribution on classes as the training set and therefore contains only canonical images. An auxiliary testing set contains all the possible images.

4.1 Nonconformity Measure

For convenience we use one of the simplest NCM that can be applied. As the space is discrete, NCM of an example with respect to the set of another ones is defined as the number of 'zero distance other class neighbors', i.e. number of examples in the set that have the same features but in fact belong to another class:

$$\alpha_i = A\left(\{(x_1,y_1),\ldots,(x_l,y_l),(x_{l+1},y)\},(x_i,y_i)\right) = card\{j : x_i = x_j, y_i \neq y_j\}.$$

4.2 Probabilistic Values of the Features

We apply our approach in its form mentioned in Section 3.3. All the features of X are binary and we assumed that the mistakes in features are done independently of each other. The connection between a 'hidden' vector

$$x = (x(1),\ldots,x(7)) \in X = \{0,1\}^7$$

and a corresponding 'visible' distribution H is the following. H is a distribution on X such that:

- $x(1),\ldots,x(7)$ are H-independent on each other;
- for each $j = 1,\ldots,7$, $H\{x(j) = 1\}$ is either $1-\gamma$ or γ;
- a mistake $x(j)$ in a feature $x(j)$ is done with probability γ;
- if there is no mistake in $x(j)$ then $H\{x(j) = 1\} = 1-\gamma$ if $x(j) = 1$, γ if $x(j) = 0$;
- if there is a mistake in $x(j)$ then $H\{x(j) = 1\} = \gamma$ if $x(j) = 1$, $1-\gamma$ if $x(j) = 0$.

This means that in the 'visible' features vectors all the features are probabilistic. Each of the features is either *1 with probability* $1-\gamma$*, 0 with probability* γ or *1 with probability* γ*, 0 with probability* $1-\gamma$.

4.3 Other Details

Following 3.2 we set the 'allowance' coefficient to $s = 0.01$. Following the note 3.4 we avoid scanning all possible combinations by calculating p-values as Monte-Carlo approximations. the number of trials is 1000. Further we will see that this approximation does not affect validity properties.

Summarizing, there were 1000 trials (i.e. random filling of the missing values), and consider as the approximate credibility p_{cred} the 10-th largest of these p-values plus the allowance $s = 0.01$.

5 Results

Remind that p_{cred} finally is the p-value assigned to a new example (x_{l+1}, y_{l+1}).

To check the validity we wish to check what p-value is assigned to the true hypothesis about y_{l+1}. The correspoding p_{cred} is called p_{true}.

If y_{l+1} is unknown then each possible hypothesis about its value should be checked and assigned a p-value. As well as in the standard conformal predictor, the *prediction* is the hypothesis with the largest p-value and *confidence* in it is 1 minus the second largest p-value.

5.1 Validity

According to our problem statement, the validity is checked on testing examples that do not contain uncertainty and have the same distribution as the training examples *before* introducing mistakes. Therefore, each of the testing examples is one of ten digits ($y \in \{0,1,\ldots,9\}$) presented with its canonical image x. In order to satisfy i.i.d. assumption with training set, the distribution of ten types also corresponds to one from USPS data.

The corresponding validity plot is presented on Fig.2. It show that the probability of error (true value being outside the prediction set) does not exceed the selected significance level, for example:

$$P\{p_{true} \leq 0.16\} = 0.08;$$

$$P\{p_{true} \leq 0.27\} = 0.17.$$

The validity is satisfied with some excess. The same effect is known for the standard credibility and for LUPI due to involving incomplete information into the data.

5.2 Confidence

Recall that the testing set consists only of canonical images, so there are only 10 possible different configurations.

Individual confidences for them can be seen on Fig. 5.2 (boxed items), average value is 0.87. The smallest of these confidences is 0.79 assigned to the digit 7, because this digit is mixable with 1 (Hamming distance between them is the smallest) and relatively rare in the training set.

For comparison we also included confidence values that would be assigned to all 128 possible pixel combinations (auxiliary testing set) and they are much lower (0.18 in average).

The more indefinite the data are the smaller is the achieved level of confidence. For example, if we increase the probability of mistake from $p_0 = 0.1$ to $p_0 = 0.2$ then the figures of average confidence falls down to 0.55 and 0.11 respectively.

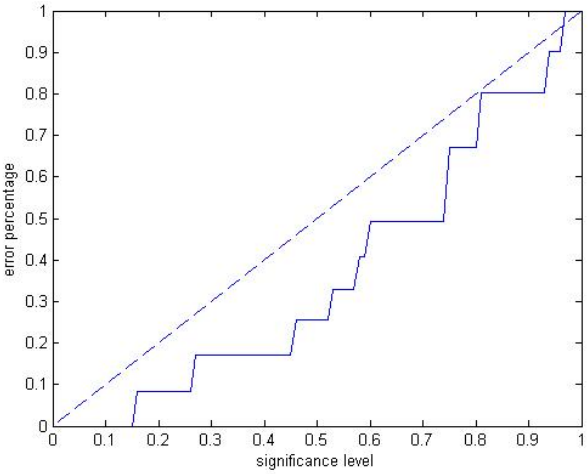


Fig. 2. Validity plot

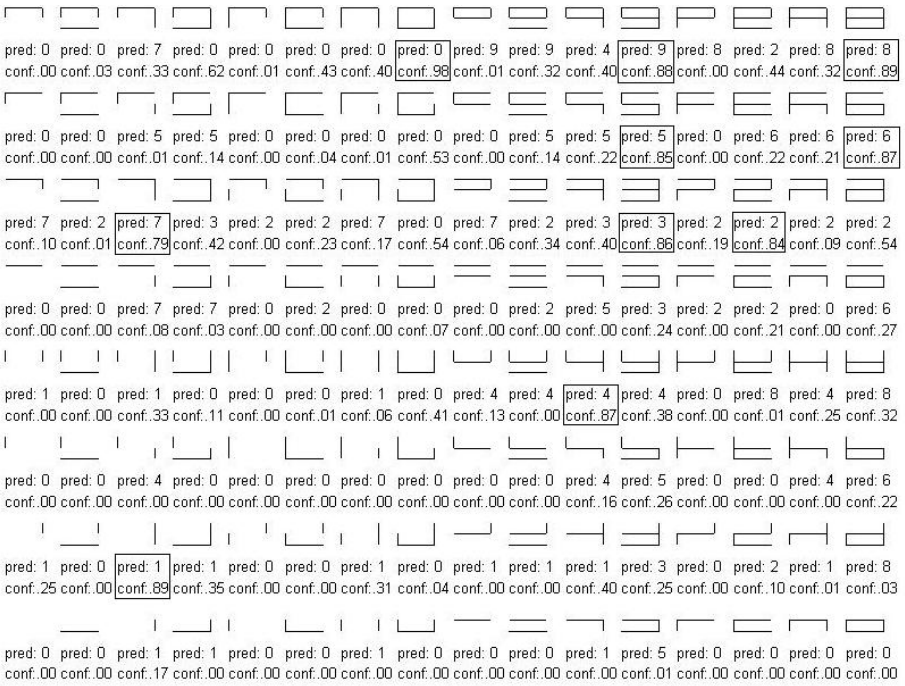


Fig. 3. Predictions for all possible pixel combinations. Predictions for the canonical images are put in boxes.

6 Conclusion

In this work we formulated an approach to get confident prediction from the data with missing values (or labels) presented in a probabilistic form. Probabilistic input means that there is an a priori distribution on possible filling of these missing values.

The advantages of conformal approach for this task are not ignoring examples with incomplete information, and on the other hand not wasting time attempting to restore the missing values.

The missing features are taken as a priori distributions on their possible values. This is an analogue of Bayesian distribution on a parameter of a statistical model. So we can expect as well that it might be assumed in other practical problems with incomplete information.

Acknowledgments. This work was supported by EPSRC grant EP/K033344/1 ("Mining the Network Behaviour of Bots"); by Thales grant ("Development of automated methods for detection of anomalous behaviour"); by the National Natural Science Foundation of China (No.61128003) grant; and by grant 'Development of New Venn Prediction Methods for Osteoporosis Risk Assessment' from the Cyprus Research Promotion Foundation.

We are grateful to Judith Klein-Seetharaman and Alex Gammerman for motivating discussions.

References

1. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer (2005)
2. LED Display Domain Data Set, `http://archive.ics.uci.edu/ml/datasets/LED+Display+Domain`
3. Fedorova, V., Gammerman, A., Nouretdinov, I., Vovk, V.: Conformal prediction under hypergraphical models. In: Papadopoulos, H., Andreou, A.S., Iliadis, L., Maglogiannis, I. (eds.) AIAI 2013. IFIP AICT, vol. 412, pp. 371–383. Springer, Heidelberg (2013)
4. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Netwotks 22, 544–557 (2009)
5. Yang, M., Nouretdinov, I., Luo, Z.: Learning by Conformal Predictors with Additional Information. In: Papadopoulos, H., Andreou, A.S., Iliadis, L., Maglogiannis, I. (eds.) AIAI 2013. IFIP AICT, vol. 412, pp. 394–400. Springer, Heidelberg (2013)
6. Adamskiy, D., Nouretdinov, I., Gammerman, A.: Conformal prediction in semi-supervised case. In: Post-Symposium Book 'Statistical learning and Data Science'. Chapman and Hall, Paris (2011)
7. Nouretdinov, I., Gammerman, A., Qi, Y., Klein-Seetharaman, J.: Determining Confidence of Predicted Interactions Between HIV-1 and Human Proteins Using Conformal Method. In: Pacific Symposium on Biocomputing, vol. 17, pp. 311–322 (2012)
8. Lei, J., Robins, J., Wasserman, L.: Efficient Nonparametric Conformal Pre- diction Regions. arXiv:1111.1418v1