Reverse conformal approach for on-line experimental design

Ilia Nouretdinov

Computer Learning Research Centre
Royal Holloway*, University of London

June 14, 2017

1. Conformal prediction
2. Reverse (object-by-label) conformal prediction
3. Conformal prediction for transfer learning
4. Experimental design methodology
5. Experimental running

Experimental design problem

This work is motivated by an experimental design problem which is likely to appear in such areas as drug design. Assume that there is a set of instances (e.g. chemical compounds) and the task is to find an item with a desired property. For any of the items, this can be done by experimental validation but this is costly. The success (reward) is the percentage of experiments which were successful in the sense that the selected object had really shown to have the desired property. On-line setting assumes that after selecting an instance, a natural experiment makes its label known for further research.

Conformal prediction

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 $x_1, x_2, \ldots, x_l \in X$ supplied with labels $y_1, y_2, \ldots, y_l \in Y$.
The core detail of conformal predictor for a *non-conformity measure (NCM) A* that is a measure of information distance an object z, which is usually a labelled feature vector, and a set U of objects of the same nature. In the case of supervised learning, a p-value is assigned to each possible hypothesis $y \in Y$ about the label of the new object $x_{l+1} \in X$:

$$p(y) = p(z_1, \ldots, z_l, (x_{l+1}, y)).$$

calculated as

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

The prediction set $R^{\varepsilon} \subset Y$ is the set of y s.t.

$$p(y) = p\left(z_1, \ldots, z_l, (x_{l+1}, y)\right) > \varepsilon.$$

Reverse (object-by-label) conformal prediction

Assume now that the problem is the opposite: to guess which object x should have a desired label y. For this problem we need to select a *testing set* S which may the whole space X or its subset.
Generally, the x-prediction set $R_h^\varepsilon \subset S \subset X$ is the set of $x \in S$ s.t.

$$p_h(x) = p\left(z_1, \ldots, z_l, (x, h)\right) > \varepsilon.$$

The validity property for x-prediction set may be understood in the following way. Assume that we are looking for an example labelled $y = h$ within the testing set S. If $x \in S$ does really have this property (label $y = h$), then it is covered by R_h^ε with probability at least $1 - \varepsilon$.
Efficiency of the prediction can be measured by small size of the prediction set for given significance level ε, or by average p-value on S.

Conformal prediction for transfer learning

Conformal prediction for transfer learning is developed by Zhou, Smirnov et al.
There are two training sets instead of one: main (target) set T and addition (source) set S that may have some deviation from i.i.d.
The learning is done in the following way. Let $T = \{z_1, \ldots, z_t\}$, $S = \{z_{t+1}, \ldots, z_l\}$; non-conformity scores can be defined as before:

$$\alpha_i = A\left(z_i, \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_{l+1}\}\right).$$

However calculation of p-values is changed:

$$p = \frac{card\{i = 1, 2, \ldots, t, l+1 : \alpha_i \geq \alpha_{l+1}\}}{t+1}.$$

Experimental running

As a toy example, we use the Mushroom data set from UCI repository mushroom. The data set contains 8,124 instances with 22 discrete attributes and 2 classes (whether a mushroom is edible or poisonous). 4,208 instances blog to the positive (edible) class.
The non-conformity score of an object is defined as 1NN Hamming distance to the nearest same class object divided by Hamming distance to the nearest other class object.
We assume that the number of possible experiments is 100, which is the sum of random and active phases of the learning:

$$A + B = 100.$$

The efficiency is measure by percentage of positive (edible) examples found during both phases of learning together.

Algorithm 1 Selection by largest p-value

A and B are lengths of random and active phases
start with a randomly chosen $K_0 \subset \{1, \ldots, N\}$ of size A
FOR $t := 1$ TO B
*** The suggested rule of choice for j_t ***
create the candidate set: $C_t = F(K_{t-1})$
FOR $c \in C_t$
train CP on K_0 with transfer from $K_{t-1} \setminus K_0$
test on x_c
assign p-value p_c to the hypothesis that $y_c = 1$
ENDFOR
*** Select the candidate with the largest p-value ***
$j_t = \arg\max_{c \in C_t} p_c$
$K_t = K_{t-1} \cup \{j_t\}$.
*** Now y_{j_t} becomes open ***
ENDFOR

Algorithm 2 Selection by largest reward

start with $K_0 \subset \{1, \ldots, N\}$
FOR $t := 1$ TO B
create the candidate set: $C_t = F(K_{t-1})$
create a randomly selected testing set: $S \subset \{1, \ldots, N\} \setminus (K_{t-1} \cup C_t)$
FOR $c \in C_t$
train CP on K_0 with transfer from $K_{t-1} \setminus K_0$
test on x_c; assign p_c to the hypothesis $y_c = 1$
END FOR
create the short list: $C'_t = \{c \in C_t : p_c > \varepsilon\}\}$
FOR $c \in C'_t$
train CP on K_0 with transfer from $(K_{t-1} \cup \{c\}) \setminus K_0$ (with $y_c = 0$)
test on S
measure the reward $r_c = card\{s \in S : p_s < \epsilon\}$ where p_s is for $y_s = 1$
ENDFOR
$j_t = \arg\max_{c \in C'_t} r_c$
$K_t = K_{t-1} \cup \{j_t\}$.
ENDFOR

Algorithm 1 results

In the table $\phi < 1$ means that the positive (edible) class is reduced.
The results are averaged over 100 random seeds.

ϕ	A	B	Reward
1	2	98	62.72
1	6	94	85.80
1	10	90	88.58
1	12	88	88.80
1	14	86	88.85
1	16	84	**88.97!**
1	18	82	88.75
1	20	80	88.52
1	30	70	85.39
1	40	60	81.28
1	50	50	76.34
1	70	30	66.70
1	100	0	52.20

ϕ	Reward
0.2	29.18
0.2	41.77
0.2	45.29
0.2	**45.89!**
0.2	45.01
0.2	43.71
0.2	44.39
0.2	43.06
0.2	38.08
0.2	34.52
0.2	30.74
0.2	26.31
0.2	17.80

ϕ	A	B	Reward
0.1	2	98	14.88
0.1	6	94	22.94
0.1	10	90	25.16
0.1	12	88	25.46
0.1	14	86	26.38
0.1	16	84	**26.66!**
0.1	18	82	26.50
0.1	20	80	25.36
0.1	30	70	23.09
0.1	40	60	20.42
0.1	50	50	18.09
0.1	70	30	14.52
0.1	100	0	9.62

Algorithm 1 vs Algorithm 2

ϕ	A	B	Algorithm 1 (Sec. alg1)	Algorithm 2 (Sec. res2) $\epsilon = 0.05$	$\epsilon = 0.01$	$\epsilon = 0.001$
0.1	2	98	14.88	15.39	15.39	15.39
0.1	6	94	22.94	26.50	26.50	26.50
0.1	10	90	25.16	31.38	31.38	31.38
0.1	12	88	25.46	33.15	33.15	33.15
0.1	14	86	26.38	35.47	35.47	35.47
0.1	16	84	26.66!	36.18 !	36.18	36.18
0.1	18	82	26.50	15.37	36.40 !	36.43 !
0.1	20	80	25.36	22.62	36.07	36.07
0.1	30	70	23.09	20.64	34.93	34.93
0.1	40	60	20.42	18.92	32.45	32.45
0.1	50	50	18.09	16.67	28.72	28.72
0.1	70	30	14.52	13.49	21.21	21.21
0.1	100	0	9.62			

Table : Results for Algorithms 1 and 2

Conclusion

This paper have shown how the experimental design can be done on the based on the conformal prediction, and how the validity of conformal prediction can be saved by means of transfer learning. Although conformal prediction is based on i.i.d. assumption and therefore requires some part of the data for the experiments to be selected fairly randomly, this part is usually not a large one ($12 - 16\%$) as shown in the experiments.
However, this is a toy example, and there are question for future research. The main of them is how to increase the achieved quality further. Application of Algorithm 2 shows a possible direction of improvement, requiring more studies. The results are promising although the method is more time-consuming.