# conformalClassification: A Conformal Prediction R Package for Classification

*by Niharika Gauraha and Ola Spjuth*

**Abstract** The conformalClassification package implements Transductive Conformal Prediction (TCP) and Inductive Conformal Prediction (ICP) for classification problems. Conformal Prediction (CP) is a framework that complements the predictions of machine learning algorithms with reliable measures of confidence. TCP gives results with higher validity than ICP, however ICP is computationally faster than TCP. The package conformalClassification is built upon the random forest method, where votes of the random forest for each class are considered as the conformity scores for each data point. Although the main aim of the conformalClassification package is to generate CP errors (p-values) for classification problems, the package also implements various diagnostic measures such as deviation from validity, error rate, efficiency, observed fuzziness and calibration plots. In future releases, we plan to extend the package to use other machine learning algorithms, (e.g. support vector machines) for model fitting.

## Introduction

Conformal predictors are confidence predictors that result in prediction sets for all confidence levels. Thus, Conformal Prediction (CP) is a framework that complements the predictions of machine learning algorithms with reliable measures of confidence. Transductive Conformal Prediction (TCP) works in an on-line transductive setting, such that learning and prediction occur simultaneously. In this sense confidence in a prediction is tailored both to the previously seen objects (whose features and labels are known) and to the features of the new object, whose label is to be predicted. By conditioning on the new objects conformal predictors take account of how difficult a particular object is to label and adjust their confidence in the prediction accordingly, as opposed to having an overall error rate for labelling all new objects Vovk et al. (2005). The output from a TCP algorithm is thus a point prediction and a region prediction, such as a 95% prediction region which, under minimal assumptions, contains the true label with a probability of at least 0.95 Shafer and Vovk (2008). The method for point prediction embedded within the CP framework can be almost any machine learning algorithm, such as random forests, support vector machines or neural networks. Based on the chosen learning algorithm a nonconformity measure is created which evaluates the “strangeness" of the new object relative to those previously seen. The TCP algorithm utilizes this nonconformity score to define the appropriate prediction region Shafer and Vovk (2008).

The fully on-line mode of TCP can be very computationally demanding (with the learning algorithm updated for each new data point). The theory however extends easily to the off-line inductive (batch) mode giving rise to what we refer to here as Inductive Conformal Prediction (ICP). CP has been used in moderately sized problems, e.g. to predict quantitative structure-activity relationships of molecules Norinder et al. (2014), to assess complication risks following coronary procedures Balasubramanian et al. (2014) and to detect anomalies in fishing vessel trajectories Smith et al. (2015). It has also been shown to scale up well on a distributed computing implementation to very large datasets, such as the Higgs boson dataset of 11 million data points Capuccini et al. (2015), the largest binary classification dataset in the UCI machine learning repository Bache and Lichman (2013).

In the release version of conformalClassification we use random forests as the underlying machine learning method, where the vote for each class – the ratio between the number of trees in the forest voting for a given class divided by the total number of tree – gives the conformity score for each data point.

## Background and Notations

This section gives a brief background about CP and fixes notations and definitions used throughout the article.

The object space is denoted by $\mathcal{X} \in \mathbb{R}^p$, where $p$ is the number of features, and label space is denoted by $\mathcal{Y} \in (1, 2, ..., l)$, where $l$ is the number of class labels. We assume that each observation consists of an object and its label, and its space is given as $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. The typical classification problem is, given a training dataset $Z = \{z_1, ..., z_n\}$ – where $n$ is the number of observations in the training set, and each observation $z_i = (x_i, y_i)$ is a labeled observation – we want to predict the label of a new observation $x_{new}$ whose label is unknown. The exchangeability (Shafer and Vovk, 2008) of

observations is assumed throughout the paper.

The nonconformity measure is a function that measures the disagreement of possible labels of a test object with respect to an observed distribution.

**Definition 2.2.1 (Nonconformity Measure)** *A nonconformity measure is a measurable function* $\mathcal{A} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ *such that* $\mathcal{A}(Z_1, Z_2)$ *does not depend on the ordering of observations in the set* $Z_1$.

The nonconformity scores are most often derived from the underlying algorithms used for point prediction. For classification problems, the error rate may be higher in some classes than others, to overcome this issue the nonconformity scores are applied on per class basis, this is referred to as Mondrian CP Norinder et al. (2014). Alternatively, the conformity measure can be defined as, $1 - \mathcal{A}$(nonconformity measure).

A natural conformity measure for classification problems using random forests method Breiman (2001) is the proportion of votes for each class, the ratio between the number of trees in the forest voting for a given class divided by the total number of trees.

$$\alpha_i(y) = \frac{\text{\#trees voting for class y}}{\text{\#of trees}} \tag{1}$$

We denote by $\alpha_i(y)$, the conformity score for $i^{th}$ observation for class $y$. Each component $\alpha_i(y)$ that corresponds to the sample $(x_i, y_i)$ is computed by equation (1) based on the augmented sample $\{z_1, ..., z_n, z_{n+1} = (x_{new}, y)\}$. Then p-value as defined below, Vovk et al. (2005), describes the lack of conformity of the new observation $x_{new}$ to the training set $Z$.

$$p_y = \frac{|\{z_i \in Z : y_i = y, \alpha_i(y) < \alpha_{new}(y)\}| + u_i * |\{z_i \in Z : y_i = y, \alpha_i(y) = \alpha_{new}(y)\}|}{n_y + 1}$$

where $u_i \sim U[0,1]$, $n_y$ denotes the number of observations having the true label as class-y in the training set. The p-value $p(y) = p_y,\ y \in Y$ lies in $\left(\frac{1}{n_y+1}, 1\right)$. The smaller the $p(y)$ is, the less likely the true pair is $(x_{new}, y)$. Multiplying the borderline cases by $u_i$ results in what are known as smoothed conformal predictors (Vovk et al., 2005).

**Definition 2.2.2 (Transductive Conformity Prediction (TCP))** *Given a training dataset* $Z$ *and a new observation* $x_{new}$*, the transductive conformal predictor (TCP ), corresponding to a nonconformity measure* $\mathcal{A}$*, checks each of a set of hypothesis (for all possible labels) for the new observation* $x_{new}$*, assigns to it a p-value, and finds the prediction region for the test set* $x_{new}$ *at a significance level* $\epsilon \in (0, 1)$.

The predicted region of a test observation is a subset of $\mathcal{Y}$ , denoted as $\Gamma^\epsilon = \{y \mid p_y > \epsilon\}$, at a significance level $\epsilon \in (0, 1)$. A prediction region $\Gamma^\epsilon = \{y \mid p_y > \epsilon\}$ contains the true value of a test observation with probability at least $1 - \epsilon$. The prediction region $\Gamma^\epsilon$ can be any one of the following:

1. Empty, when $|\Gamma^\epsilon| = 0$.
2. Singleton, when $|\Gamma^\epsilon| = 1$.
3. Multiple, when $|\Gamma^\epsilon| > 1$.

**Algorithm 1: TCP**

**Input:** (training dataset:$Z$, test data:$x_{new}$, label set:$Y$, a nonconformity measure:$\mathcal{A}$)
**Output: p-values**
**for** *each* $y \in \mathcal{Y}$ **do**
  $z_{n+1} = (x_{new}, y)$;
  $Z^* = (Z, z_{n+1})$ ;
  Compute the transductive nonconformity scores:
  $\alpha_i = \mathcal{A}(Z^*, z_i)$ for each $z_i \in Z^*$;
  Compute p-value: $p(y) = \frac{|\{i \in \{1,..,n+1\} : y_i = y, \alpha_i(y) < \alpha_{new}(y)\}| + u_i * |\{i \in \{1,..,n+1\} : y_i = y, \alpha_i(y) = \alpha_{new}(y)\}|}{n_y + 1}$;
**end**
**p-values** $= \{p(y) | y \in \mathcal{Y}\}$;
**return p-values**;

For further details on TCP, we refer to Vapnik and Vapnik (1998), Shafer and Vovk (2008), Vovk et al. (2005) and Balasubramanian et al. (2014).

The computational expense of TCP, whereby the prediction rule is updated for each new example for each class label, may be computationally intractable for large datasets. To address this issue the batch-mode ICP method was introduced. For ICP, the training set $Z$ is partitioned into two different sets: the proper training set, $Z_p = z1, ..., z_q$ of size $q$, and the calibration set $Z_c = z_{q+1}, ..., z_n$ of size $n - q$. The validity of ICP depends upon how well the calibration set conforms to the proper training set. The ICP p-value is then computed as

$$p_y = \frac{|\{z_i \in Z_c : y_i = y, \alpha_i(y) < \alpha_{new}(y)\}| + u_i * |\{z_i \in Z_c : y_i = y, \alpha_i(y) = \alpha_{new}(y)\}|}{n_y + 1},$$

where $n_y$ denotes the number of observations having the true label as class-y in the calibration set.

**Algorithm 2: ICP**

**Input:** (training dataset:$Z$, test data:$x_{new}$, label set:$Y$, a nonconformity measure:$\mathcal{A}$)
**Output: p-values**
partition $Z$ into proper training set $Z_p$ and calibration set $Z_c$
Compute nonconformity scores:
$\alpha_i = \mathcal{A}(Z_p, z_i)$ for each $z_i \in Z_c$;

Compute nonconformity score for test observation: $\alpha_{new} = \mathcal{A}(Z_p, (x_{new}, y))$ for each $y \in Y$
Compute p-values:
$p(y) = \frac{|\{z_i \in Z_c : y_i = y, \alpha_i(y) < \alpha_{new}(y)\}| + u_i * |\{z_i \in Z_c : y_i = y, \alpha_i(y) = \alpha_{new}(y)\}|}{n_y + 1}$;
**p-values** $= \{p(y) | y \in Y\}$;
**return p-values**;

To evaluate the performance of conformal predictors, we consider the following criterion: error rate, validity, efficiency and observed fuzziness. A predictor makes an error when the predicted region does not contain the true label, that is $y \notin |\Gamma^\epsilon|$. Given a training dataset $Z$ and an external test set $Z_T$, and $|Z_T| = m$. Suppose that a conformal predictor gives prediction regions as $\Gamma^\epsilon_1, ...., \Gamma^\epsilon_m$, then the error rate is defined as follows.

**Definition 2.2.3 (Error rate)**

$$ER^\epsilon = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{I}_{\{y_i \notin \Gamma^\epsilon_i\}}, \tag{2}$$

*where $y_i$ is the true class label of the $i^{th}$ test case and $\boldsymbol{I}$ is an indicator function.*

The efficiency can be computed as the ratio of predictions with more than one class over number of observations in the test set.

**Definition 2.2.4 (Efficiency)**

$$EFF^\epsilon = \frac{1}{m} \sum_{i=1}^{k} I_{(|\Gamma^\epsilon| > 1)} \tag{3}$$

The deviation from exact validity can be computed as (Carlsson et al. (2017)) the Euclidean norm of the difference of the observed error and the expected error for a given set of predefined significance levels. Let us assume a set of significance levels $\epsilon = \{\epsilon_1, ..., \epsilon_k\}$, then the formula for the validity can be given as follows.

**Definition 2.2.5 (Deviation from Validity)**

$$VAL = \sqrt{\sum_{i=1}^{k} (ER^{\epsilon_i} - \epsilon_i)^2} \tag{4}$$

The Observed fuzziness is defined as the sum of all p-values for the incorrect class labels.

**Definition 2.2.6 (Observed Fuzziness)**

$$ObsFuzz = \frac{1}{m} \sum_{i=1}^{m} \sum_{y_i \neq y} p_i^y, \tag{5}$$

We note that for the above measure of performances, smaller values are preferable.

## Conclusions

The conformalClassification package implements Transductive Conformal Prediction and Inductive Conformal Prediction for Classification problems using Random Forests as the underlying machine learning algorithm.

## Future Development

In future releases, we plan to extend package to use other machine learning algorithms, (e.g. support vector machines) for model fitting.

## Acknowledgements

The authors acknowledge UPPMAX, Uppsala Multidisciplinary Centre for Advanced Computational Science for providing computational resources. The authors would also like to thank Philip J. Harrison for comments and recommendations during the preparation of this manuscript and R package.

## Bibliography

K. Bache and M. Lichman. Uci machine learning repository. 2013. [p1]

V. Balasubramanian, S.-S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Newnes, 2014. [p1, 2]

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324. [p2]

M. Capuccini, L. Carlsson, U. Norinder, and O. Spjuth. Conformal prediction in spark: Large-scale machine learning with confidence. In *Big Data Computing (BDC), 2015 IEEE/ACM 2nd International Symposium on*, pages 61–67. IEEE, 2015. [p1]

L. Carlsson, C. Bendtsen, and E. Ahlberg. Comparing performance of different inductive and transductive conformal predictors relevant to drug discovery. In *Conformal and Probabilistic Prediction and Applications*, pages 201–212, 2017. [p3]

U. Norinder, L. Carlsson, S. Boyer, and M. Eklund. Introducing conformal prediction in predictive modeling. a transparent and flexible alternative to applicability domain determination. *Journal of chemical information and modeling*, 54(6):1596–1603, 2014. [p1, 2]

G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar): 371–421, 2008. [p1, 2]

J. Smith, I. Nouretdinov, R. Craddock, C. Offer, and A. Gammerman. Conformal anomaly detection of trajectories with a multi-class hierarchy. In *International Symposium on Statistical Learning and Data Sciences*, pages 281–290. Springer, 2015. [p1]

V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998. [p2]

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005. [p1, 2]

*Niharika Gauraha*
*Uppsala University*
*Uppsala*
*Sweden*
niharika.gauraha@farmbio.uu.se

*Ola Spjuth*
*Uppsala University*
*Uppsala*
*Sweden*
ola.spjuth@farmbio.uu.se