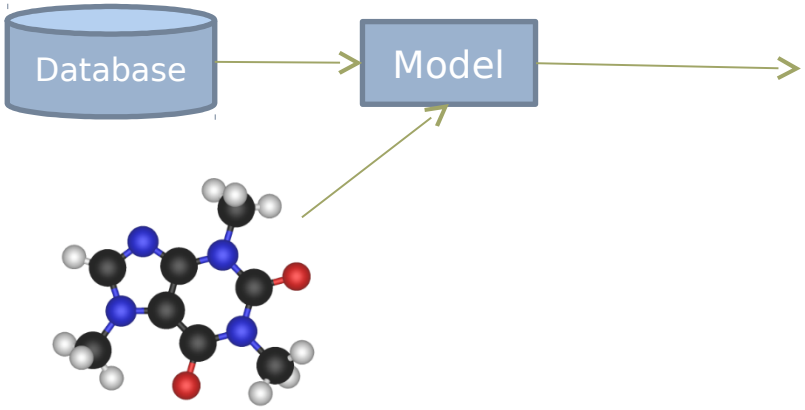
Conformal prediction

Henrik Boström
Dept. of Computer and Systems Sciences,
Stockholm University & RISE SICS
henrik.bostrom@{dsv.su.se,ri.se}

- Standard vs. conformal prediction
- Conformal classification
- Conformal regression
- Conformal prediction with trees and forests

Standard prediction

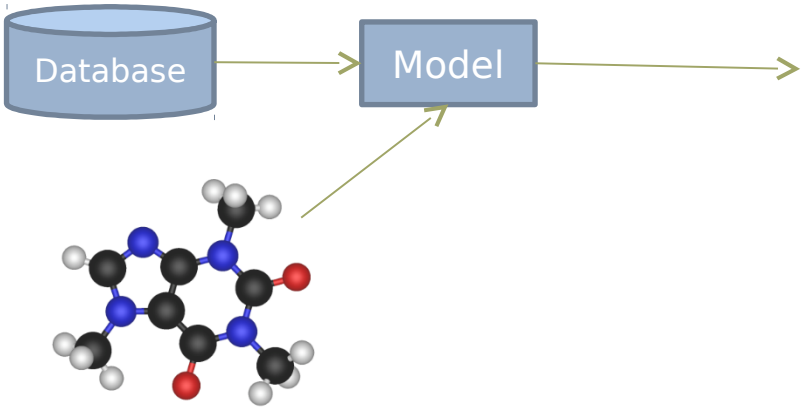


Classification
y' = toxic
P(y' = y) = ?

Regression
y' = 5.2
P(y' = y) = ?

- We can measure the performance of a model, but how certain are we about specific predictions?
- What if we would need to control the error level?

Conformal prediction



Classification
$P(y \in Y') = 0.95$
$Y' \subseteq \{toxic, nontoxic\}$

Regression
$P(y \in Y') = 0.95$
$Y' = (4.9,5.5)$

- Guaranteed error-level (1-confidence)
- Set-valued predictions (size reflects uncertainty)
- Can be coupled with any underlying predictive model

V. Vovk, A. Gammerman and G. Shafer.
Algorithmic learning in a random world. Springer. 2006.

Stockholm University

Conformal prediction - how does it work?

In the inductive (batch) setting, we are given:

1) **a predictive model** *h* [from your favorite algorithm]

2) **a calibration set** (not used for building *h*)

3) **a non-conformity measure** *A*, which returns a score *A(x,y,h),* given instance *x* and label *y*

Stockholm University

The non-conformity measure

The non-conformity measure should capture the *strangeness* of the label *y* for the instance *x*.

Some commonly used functions:

$A(x,y,h)=1-P_h(y,x)$ [for classification]

$A(x,y,h)=|y-h(x)|$ [for regression]

Note: any function will do for classification - including rand()!

Conformal classification

- Calculate the calibration scores

$\alpha_1 = A(x_1,y_1=nontoxic,h) = 1\text{-}P_h(nontoxic,x_1) = 0.10$
$\alpha_2 = A(x_2,y_2=toxic,h) = 1\text{-}P_h(toxic,x_2) = 0.14$
...
$\alpha_n = A(x_n,y_n=nontoxic,h) = 1\text{-}P_h(nontoxic,x_n) = 0.64$

- When making a prediction, calculate a p-value for each possible class label l_j, and exclude the label if the p-value is less than or equal to 1-c (c = confidence)

$\alpha_{n+1} = A(x_{n+1},y_{n+1}=l_j,h)$
$p_{l_j} = |\{\alpha_i : \alpha_i \geq \alpha_{n+1}\}|/(n+1)$

$\alpha_{n+1} = A(x,y'=toxic,h) = 1\text{-}P_h(toxic,x) = 0.85$ $\quad p_{toxic} = 0.03$
$\alpha_{n+1} = A(x,y'=nontoxic,h) = 1\text{-}P_h(nontoxic,x) = 0.15$ $\quad p_{nontoxic} = 0.75$

$Y' = \{nontoxic\}$

- By following this procedure, the probability of excluding the correct label is not greater than 1-c

Conformal classification

Stockholm University

- Is the error level guaranteed for each class?
  No. But this can be fixed using a separate calibration set for each class, i.e., using a *Mondrian approach*

  $\alpha_{1,1} = A(x_{1,1}, y_{1,1}=toxic, h) \ldots \alpha_{1,n_1} = A(x_{1,n_1}, y_{n,n_1}=toxic, h)$
  $\alpha_{2,1} = A(x_{2,1}, y_{2,1}=nontoxic, h) \ldots \alpha_{2,n_2} = A(x_{2,n_2}, y_{2,n_2}=nontoxic, h)$

- When making a prediction, exclude a label l_j if the p-value is less than or equal to 1-c

  $\alpha_{j,n_j+1} = A(x_{n_j+1}, y_{j\,n+1}=l_j, h)$
  $p_{l_j} = |\{\alpha_{j,i} : \alpha_{j,i} \geq \alpha_{j,n_j+1}\}|/(n_j+1)$

  $\alpha_{1,n_1+1} = A(x, y'=toxic, h) = 1\text{-}P_h(toxic, x) = 0.85$ $\quad p_{toxic} = 0.09$
  $\alpha_{2,n_2+1} = A(x, y'=nontoxic, h) = 1\text{-}P_h(nontoxic, x) = 0.15$ $\quad p_{nontoxic} = 0.83$

  $Y' = \{toxic, nontoxic\}$

- The probability of excluding the correct label is now not greater than 1-c for each class

Conformal regression

- Calculate the calibration scores

  $\alpha_1 = A(x_1,y_1=5.3,h) = |5.3-h(x_1)| = 0.10$
  $\alpha_2 = A(x_2,y_2=2.9,h) = |2.9-h(x_2)| = 0.14$
  ...
  $\alpha_n = A(x_n,y_n=4.1,h) = |4.1-h(x_n)| = 0.64$

- Let α_c be the score at the c (=confidence) percentile

  $\alpha_{0.95} = 0.55$

- When making a prediction, exclude all values with a distance greater than α_c: $Y' = (h(x)-\alpha_c, h(x)+\alpha_c)$

  $Y' = (5.2-0.55,5.2+0.55) = (4.65,5.75)$

- By following this procedure, the probability of excluding the correct label is not greater than 1-c

Conformal regression

Stockholm University

- Will we get different sizes for the predicted intervals? No. But this can be fixed, by *normalization*.

  $A(x,y,h)=|y-h(x)|/\sigma(x)$

- *Calculate the calibration scores*

  $\alpha_1 = A(x_1,y_1=5.3,h) = |5.3-h(x_1)|/\sigma(x_1) = 1.01$
  $\alpha_2 = A(x_2,y_2=2.9,h) = |2.9-h(x_2)|/\sigma(x_2) = 2.23$
  ...
  $\alpha_n = A(x_n,y_n=4.1,h) = |4.1-h(x_n)|/\sigma(x_n) = 8.74$

- Let α_c be the score at the c (=confidence) percentile

  $\alpha_{0.95} = 5.5$

- When making a prediction, exclude all values with a distance greater than $\sigma(x)\alpha_c$: $Y' = (h(x)-\sigma(x)\alpha_c, h(x)+\sigma(x)\alpha_c)$

  $Y' = (5.2-0.1*0.55, 5.2+0.1*0.55) = (4.65, 5.75)$

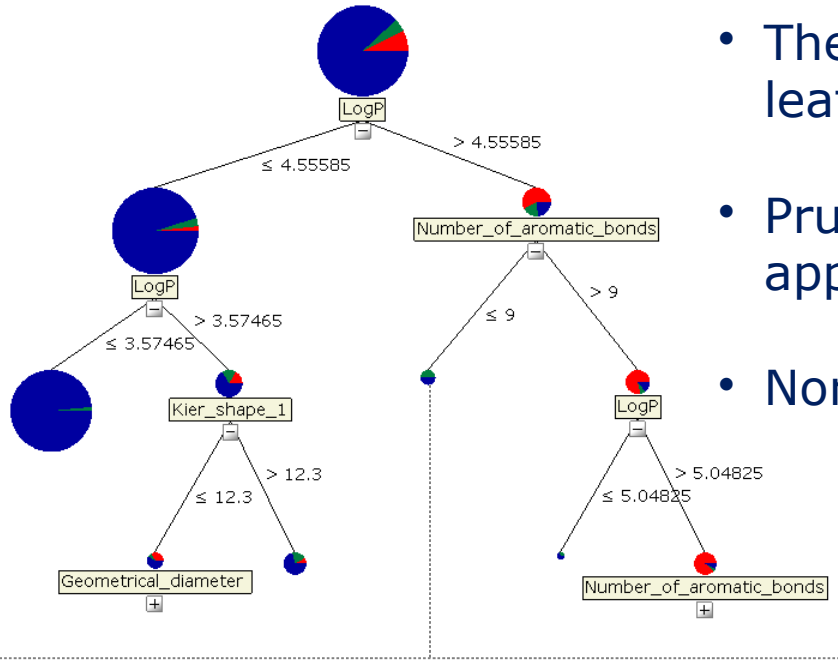
- This procedure still has the same guarantee

Stockholm University

Conformal prediction - how to evaluate it?

- Not by error level since it is determined by the user (who sets the confidence level)
- Instead, conformal predictors are evaluated with the respect to their *efficiency*, i.e., the size of the output prediction sets
- For classification, this is often measured by the average number of output labels
- For regression, this is often measured by the average size of predicted intervals

Conformal prediction for decision trees



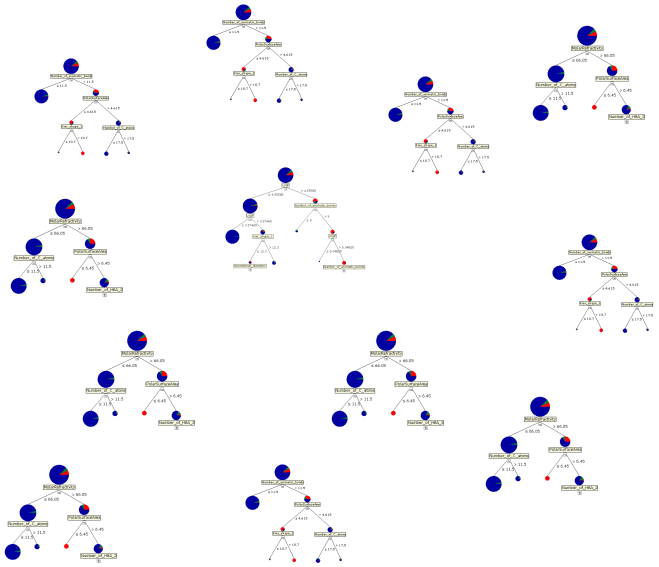
- The error-level can be guaranteed at each leaf using a Mondrian approach
- Pruning should be avoided and smoothing applied for maximum efficiency
- Normalization may affect interpretability

IF LogP > 4.56 AND Number_of_aromatic_bonds ≤ 9
THEN Y' = {GoodSolubility, MediumSolubility}

U. Johansson, H. Boström and T. Löfström. Conformal Prediction Using Decision Trees. ICDM pp. 330-339, 2013

U. Johansson, C. Sönströd, H. Linusson and H. Boström. Regression trees for streaming data with local performance guarantees. BigData, pp 461-470, 2014

Conformal prediction for random forests



- Forests have a tendency to promote the majority class, which can be addressed by a Mondrian approach
- State-of-the-art performance for conformal regression is achieved using out-of-bag instances for calibration
- Difficulty can be estimated using variance of predictions

T. Löfström, H. Boström, H. Linusson and U. Johansson. Bias reduction through conditional conformal prediction. Intelligent Data Analysis 19(6), pp. 1355-1375, 2015

U. Johansson, H. Boström, T. Löfström and H. Linusson. Regression conformal prediction with random forests. Machine Learning 97(1-2), pp. 155–176, 2014.

H. Boström, H. Linusson, T. Löfström and U. Johansson. Evaluation of a Variance-Based Nonconformity Measure for Regression Forests. Proc. of the International Symposium on Conformal and Probabilistic Prediction with Applications, Springer, pp. 75-89, 2016.

Stockholm University

Conformal regression

$$\alpha_i = A(\boldsymbol{x}_i, y_i, h) = |y_i - \hat{y}_i| = |y_i - h(\boldsymbol{x}_i)|, \tag{1}$$

$$\hat{Y}_j^{\delta} = \hat{y}_j \pm \alpha_{s(\delta)}, \tag{2}$$

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\sigma_i}\,. \tag{3}$$

$$\hat{Y}_j^{\delta} = \hat{y}_j \pm \alpha_{s(\delta)}\sigma_j\,. \tag{4}$$

Nonconformity measures

- Standard, i.e., no normalization, using OOB examples (M1) *
- Normalized using nearest neighbors (M2) *

$$\alpha_i = \frac{|y_i - \hat{y}_i|}{\mu_i + \beta} \qquad \mu_i = \frac{\sum_{n=1}^{k} o_n/d_n}{\sum_{n=1}^{k} 1/d_n}$$

- Normalized using variance (M3) **

$$\mu_i = \frac{\sum_{n=1}^{s} p_n^2}{s} - \left(\frac{\sum_{n=1}^{s} p_n}{s}\right)^2$$

* Evaluated in:

U. Johansson, H. Boström, T. Löfström and H. Linusson. Regression conformal prediction with random forests. Machine Learning 97(1-2), pp. 155–176, 2014.

** Proposed in:

H. Boström, H. Linusson, T. Löfström and U. Johansson. Evaluation of a Variance-Based Nonconformity Measure for Regression Forests. Proc. of the International Symposium on Conformal and Probabilistic Prediction with Applications, Springer, pp. 75-89, 2016.

Empirical investigation

33 datasets, 10-fold cross-validation, 500 trees

c = 0.90

	M1	M2	M3
Error	0.099	0.097	0.097
Rank	2.26	2.21	1.53
Efficiency	.226	.213	.216
Rank	2.79	1.42	1.79

c = 0.95

	M1	M2	M3
Error	0.049	0.048	0.048
Rank	2.32	2.03	1.65
Efficiency	.290	.264	.269
Rank	2.76	1.61	1.64

c = 0.99

	M1	M2	M3
Error	0.009	0.011	0.009
Rank	1.95	2.39	1.65
Efficiency	.445	.383	0.391
Rank	2.73	1.55	1.73

Computational cost

Common

	M1	M2	M3
Time	3.39	3.42	3.44
Rank	1.73	2.03	2.24

Calibration

	M1	M2	M3
Time	0.002	73.1	0.004
Rank	1.03	3.00	1.97

Application

	M1	M2	M3
Time	0.000	8.35	0.000
Rank	1.00	3.00	2.00

H. Boström, H. Linusson, T. Löfström and U. Johansson. Evaluation of a Variance-Based Nonconformity Measure for Regression Forests. Proc. of the International Symposium on Conformal and Probabilistic Prediction with Applications, Springer, pp. 75-89, 2016.

Using out-of-bag instances

Stockholm University

- Nonconformity scores from out-of-bag (oob) instances are not identically distributed to test instance scores

  $A(x_i, y_i, h_i) = |y_i - h_i(x_i)| / \sigma_{h_i}(x_i)$ *vs.* $A(x_i, y_i, h) = |y - h(x_i)| / \sigma_h(x_i)$

  $h_i \subseteq h$ *and* $|h_i| \approx 0.368|h|$

- Adjusted procedure (M4):

  For each test instance:

  1. Select one of the calibration (oob) instances randomly
  2. Use the subset of trees for which the calibration instance is oob to make a point prediction and estimate difficulty
  3. Use the remaining calibration instances, the point prediction and difficulty estimate to derive a prediction interval for the test instance

- This procedure guarantees that calibration and test scores are distributed indentically

H. Boström, H. Linusson, T. Löfström and U. Johansson. Accelerating difficulty estimation for conformal regression forests. Annals of Mathematics and Artificial Intelligence, 2017.

Empirical investigation

33 datasets, 10-fold cross-validation, 500 trees

	c = 0.90		c = 0.95		c = 0.99	
	M3	M4	M3	M4	M3	M4
Error	0.097	0.10	0.048	0.050	0.009	0.010
Rank	1.21	1.79	1.14	1.86	1.23	1.77
Efficiency	.216	.215	.269	.268	.390	.389
Rank	1.88	1.12	1.79	1.21	1.79	1.21

H. Boström, H. Linusson, T. Löfström and U. Johansson. Accelerating difficulty estimation for conformal regression forests. Annals of Mathematics and Artificial Intelligence, 2017.

Stockholm University

Conformal prediction: summary

- Provides guarantees for the error rate
- Results in set (rather than point) predictions
- Can be used with any (predictive) learning algorithm
- Design of non-conformity measure vital for informativeness
- Out-of-bag instances may be used in a sound way for calibration

Want to try it out?

Julia implementation of conformal random forests:
github.com/henrikbostrom/RandomForest

Python implementation (customized for the scikit-learn library):
github.com/donlnz/nonconformist