

Tutorial on using Conformal Predictive Systems in KNIME

INTRODUCING CONFORMAL PREDICTIVE SYSTEMS IN KNIME

Tuwe Löfström, Alexander Bondaletov, Artem Ryasik, Henrik Boström, Ulf Johansson

15.09.2023

✉ tuwe.lofstrom@ju.se

🏠 <https://ju.se/personinfo.html?sign=loftuw>



JÖNKÖPING UNIVERSITY
School of Engineering

CONTENTS

1. Introduction
 - Visual Programming
 - KNIME
 - Conformal Prediction
2. Implementation in KNIME
 - CP in KNIME
 - CPS in KNIME
3. Workflows in KNIME
 - Regular Prediction
 - CPS
 - Mondrian CPS
4. Experiments
 - Visualizing Results
 - Experimental setup
 - Results
5. Concluding discussion
 - Conclusions

Introduction



Machine learning and data science are gradually becoming more and more mainstream.

A particular set of tools offer users with the possibility to build up data science workflows using components.

- Each component represent a specific behaviour, such as data manipulation, modeling or visualization.

There are many tools that offer the possibility to build workflows using visual programming, such as RapidMiner, WEKA, Orange and KNIME.

KNIME is an end-to-end software platform for data science.

KNIME consists of two parts,

- KNIME Analytics Platform, which is an open-source software for end-to-end data analytics using visual programming
- KNIME Server, which is an enterprise software for team-based collaboration, automation, management, and deployment of data science workflows

KNIME has no-code support for most standard machine learning techniques off-the-shelf

A lot of additional functionality is available through community content, providing support for a very wide range of application areas

Conformal prediction has attracted a growing amount of scientific interest in recent years, with a predicted increase in attention onward.

However, it is not yet widely acknowledged outside academia as a natural and important tool to improve the quality of decision support systems relying on predictions.

Making it more easily accessible to a wider public, not necessarily knowledgeable in programming, is important for increased outreach.

Conformal regression defines a confidence interval as $\Gamma_j^\epsilon = h(x_j) \pm \alpha_s$, where the nonconformity function is $\alpha = |y_j - h(x_j)|$ and $s = \lfloor \epsilon(q + 1) \rfloor$. $(1 - \epsilon) \in (0, 1)$ is the user-defined confidence level.

To obtain individual bounds for each x_j , *normalized nonconformity function* can be used, which is defined as $\alpha = \frac{|y_j - h(x_j)|}{\sigma_j + \beta}$, where the quality (or difficulty) estimate σ_j represents our confidence in the prediction for y_j and β controls the sensitivity of the normalization term.

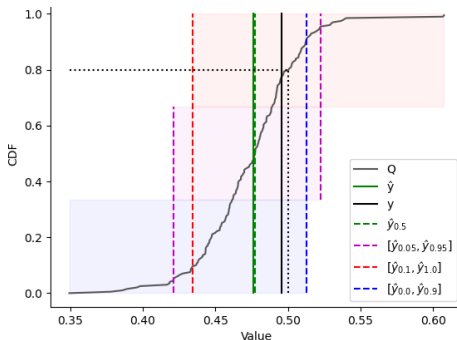
Conformal predictive systems (CPS) are an extension of conformal regressors that produce *conformal predictive distributions*, which are cumulative distribution functions.

Instead of using the absolute error, nonconformity is defined using signed errors as $\alpha = y_i - h(x_i)$ or $\alpha = \frac{y_i - h(x_i)}{\sigma_i + \beta}$ with normalization. The prediction for a test instance x_i then becomes the following cumulative conformal predictive distribution function:

$$Q(y) = \begin{cases} \frac{n+\tau}{q+1}, & \text{if } y \in (C_{(n)}, C_{(n+1)}) , & \text{for } n \in \{0, \dots, q\} \\ \frac{n'-1+(n''-n'+2)\tau}{q+1}, & \text{if } y = C_{(n)}, & \text{for } n \in \{1, \dots, q\} \end{cases} \quad (1)$$

where $C_{(i)} = h(x) + \alpha_i$ (or $C_{(i)} = h(x) + \sigma\alpha_i$ for normalization) and $C_{(1)}, \dots, C_{(q)}$ are sorted in increasing order.

Figure 1: A conformal predictive distribution with three different intervals representing 90% confidence are defined: **Lower-bounded interval**: more than the 10th percentile; **Two-sided interval**: between the 5th and the 95th percentiles; **Upper-bounded interval**: less than the 90th percentile. The black dotted lines indicate how to determine the probability of the true target being smaller than 0.5, which in this case would be approximately 80%.



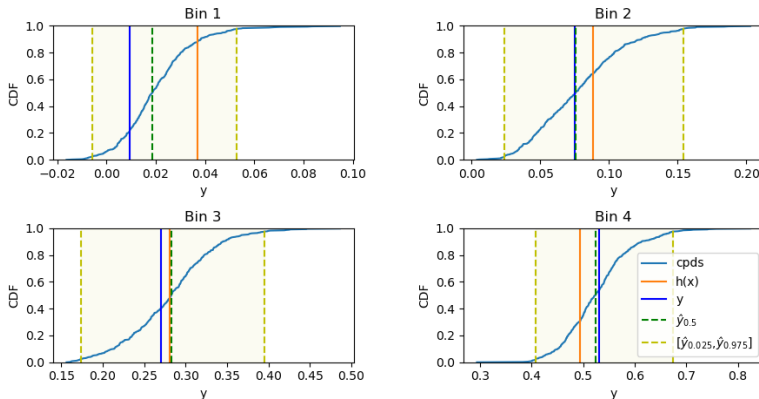
Both standard and normalized conformal predictive systems are negatively affected when the residuals are heteroscedastic.

An example of such a model is an ensemble averaging the predictions from the base regressors

- Resulting in a tendency to overestimate low actual values and underestimate high actual values.
- The distribution can only adjust for errors in one direction, as the shape of standard predictions will be the same (and σ is always positive).

Mondrian Conformal Predictive Systems divide instances based on categories and define a conformal predictive distribution only using instances in the same category.

Figure 2: Four Mondrian conformal predictive distributions from four different bins.



Implementation in KNIME



The Swedish company Redfield AB has previously developed a toolbox for conformal classification for KNIME which has been available through KNIME's official software channels as a community package.

Last year, we introduced an update and extension of the toolbox to include conformal regression as well as a number of additions to improve ease-of-use and accessibility to novice users.

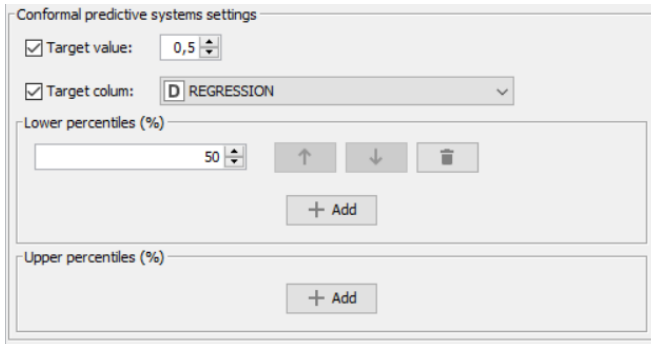
Since then, Conformal Predictive Systems has been added to the toolbox. We introduce Conformal Predictive Systems and how to set up Mondrian CPS in today's presentation.

The same node logic as previously have been used.

- Predictive Systems Calibrator (Regression) - Calculates the α -values for all calibration instances. If normalization is used, the difficulty, or σ , and β are also used in calculating α .
- Predictive Systems Predictor (Regression) - The node takes the calibration table and the test data and defines the conformal predictive distribution.

- Predictive Systems Classifier (Regression) - This node is used to define how to query the conformal predictive distribution. The node takes four types of input parameters:
 1. Target value: A fixed threshold v . When used, the node will output a column containing $\mathcal{P}(Y \leq v)$, i.e., the estimated probability that the predicted value is below v .
 2. Target Column: A column containing individual thresholds v_i for each instance.
 3. Lower Percentiles (%): A dynamic number of user-defined lower percentiles. For each lower percentile defined, the node will output a column with the values corresponding to that percentile in the conformal predictive distribution.
 4. Upper Percentiles (%): A dynamic number of user-defined upper percentiles. Works as Lower Percentiles, but for upper percentiles.

Figure 3: Selecting CPS settings



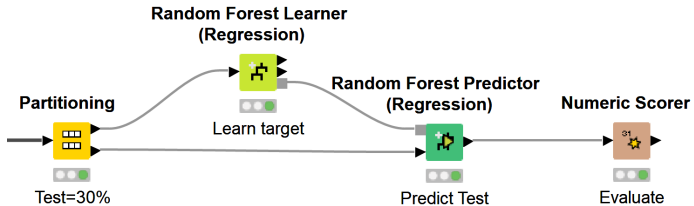
The screenshot shows a dialog box titled "Conformal predictive systems settings". It contains the following elements:

- A checked checkbox labeled "Target value:" followed by a numeric input field containing "0,5".
- A checked checkbox labeled "Target colum:" followed by a dropdown menu showing "D REGRESSION".
- A section titled "Lower percentiles (%)" containing:
 - A numeric input field with "50".
 - Three buttons: an up arrow, a down arrow, and a trash icon.
 - A "+ Add" button.
- A section titled "Upper percentiles (%)" containing a "+ Add" button.

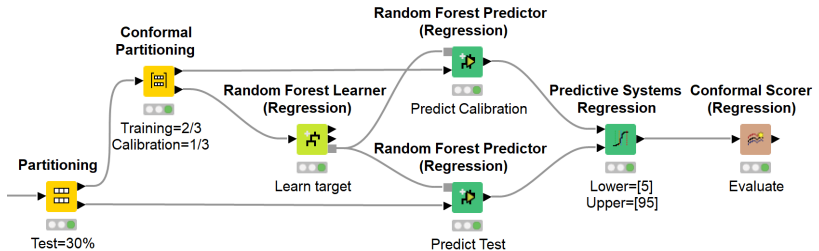
- Predictive Systems Regression - This node combines the functionality in the three nodes above with the sum of the parameters available in the individual nodes. The purpose of this node is to create an ease-of-use node for regular use cases.
- Conformal Scorer (Regression) - This node has been updated to allow evaluation of both two-sided and one-sided intervals. It compares prediction intervals with actual values and calculates metrics for estimating conformal predictions.

Workflows in KNIME





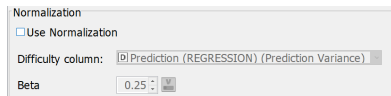
Creating a solution for conformal predictive systems is analogous to both classification and regression.



In order to use normalized conformal predictive systems, an additional column of data representing σ , i.e., how difficult an instance is, must exist in the data.

If such a column exist, changing from standard conformal predictive systems to normalized conformal predictive systems requires only opting for using normalization and selecting the σ -column.

Standard Conformal Regression



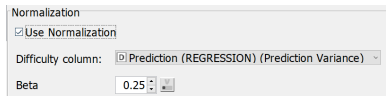
Normalization

☐ Use Normalization

Difficulty column:

Beta

Normalized Conformal Regression



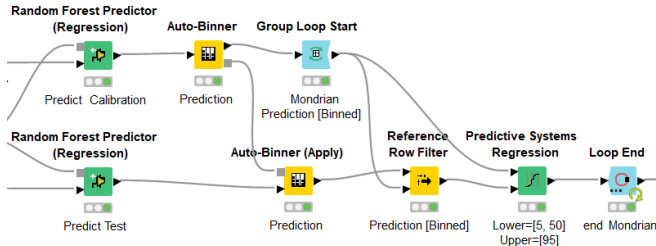
Normalization

☒ Use Normalization

Difficulty column:

Beta

Figure 4: Solution to achieve Mondrian conformal predictive systems.



Experiments





COPA 2023 - Conformal predictive systems - simple use cases

selective prediction ☐ batch ☐ regression ☐ selective prediction details ☐

Last visit Aug 01, 2023

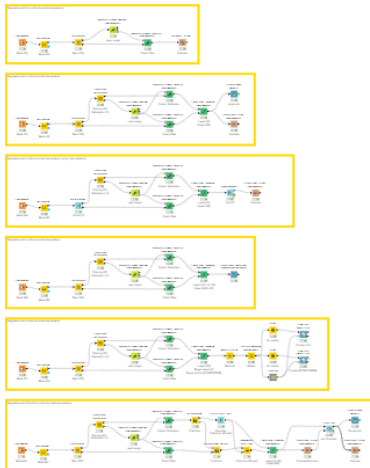
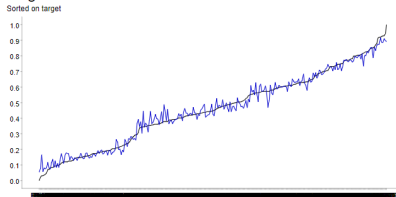
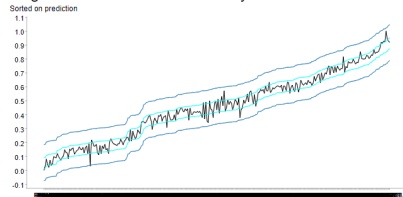


Figure 5: Standard conformal predictive systems without normalization

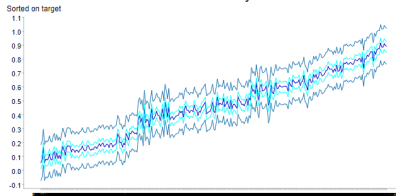
Target and Median



Target and Conformal Predictive Systems



Median and Conformal Predictive Systems



Median and Conformal Predictive Systems

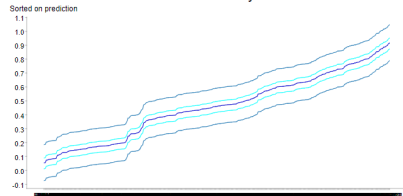
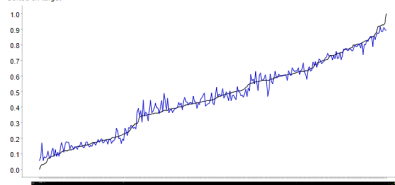


Figure 6: Normalized conformal predictive systems using the prediction variance (from a random forest) as difficulty estimate. β was set to 0.005

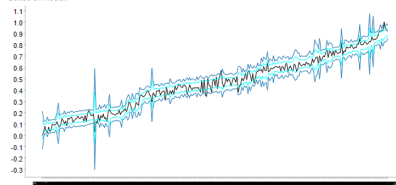
Target and Median

Sorted on target



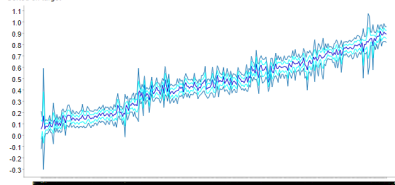
Target and Conformal Predictive Systems

Sorted on median



Median and Conformal Predictive Systems

Sorted on target



Median and Conformal Predictive Systems

Sorted on median

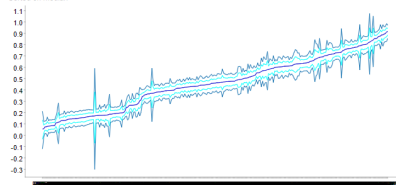


Figure 7: The target (REGRESSION), median and $P(\text{cpds} < 0.5)$, sorted on median

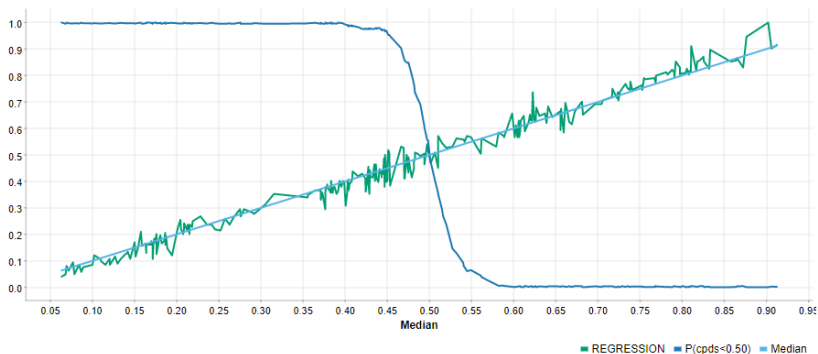
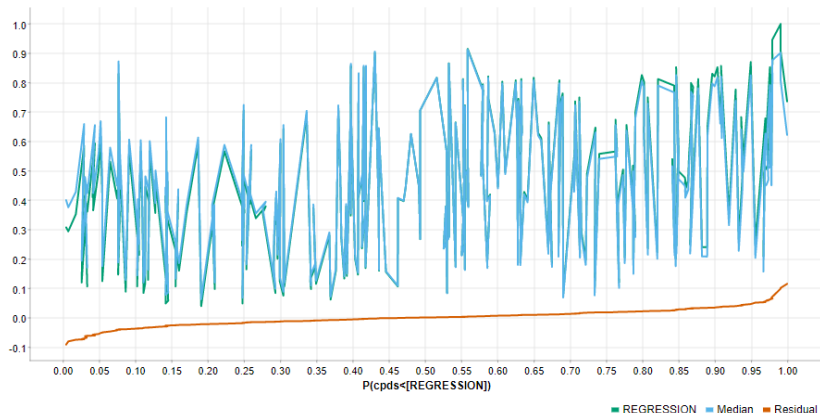


Figure 8: The target (REGRESSION), median and residual, sorted on the probability of the true target being below the prediction.



The Conformal Predictive Systems nodes have also been evaluated in experiments using 23 benchmarking data sets.

Four setups are evaluated:

- standard conformal predictive systems
- normalized conformal predictive systems using prediction variance from the random forest to estimate difficulty (σ)
- Mondrian conformal predictive systems using five equal-frequency bins defined from the prediction of the underlying model
- normalized Mondrian conformal predictive systems (combining the setups of normalized and Mondrian conformal predictive systems).

Workflow

COPA 2023 - Mondrian conformal predictive systems experiment

Mondrian Conformal predictive systems COPA Regression Uncertainty quantification +1

Last edit: Apr 8, 2023

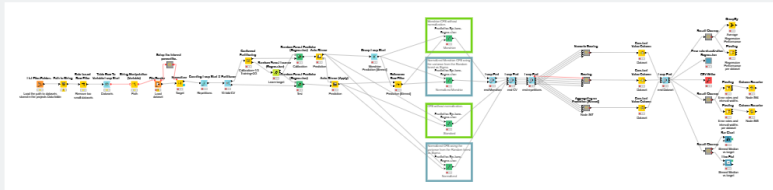
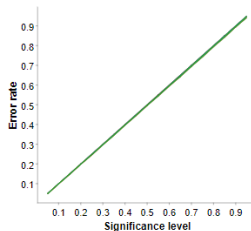
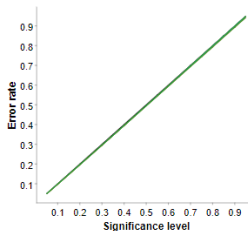


Figure 9: Error rates vs significance levels from conformal predictive systems for two-sided, lower-bounded upper-bounded intervals.

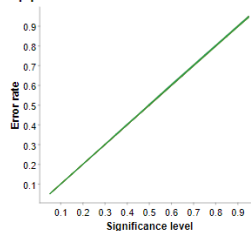
Two-sided intervals



Lower-bounded intervals



Upper-bounded intervals



■	Significance level
■	Error rate - No Normalization
■	Error rate - No Normalization Mondrian
■	Error rate - Normalization using RF variance
■	Error rate - Normalization using RF variance Mondrian

Table 1: Validity from two-sided intervals using conformal predictive systems.
 $\epsilon = 0.05$

Type of Interval	Std	N	M	NM
Two-sided	.05	.05	.05	.05
Lower-bounded	.05	.05	.05	.05
Upper-bounded	.05	.05	.05	.05

Table 2: Efficiency from two-sided intervals using conformal predictive systems.

Epsilon	Std	N	M	NM
$\epsilon = 0.05$.317	.297	.279	.279
$\epsilon = 0.10$.252	.239	.226	.223
$\epsilon = 0.20$.186	.178	.172	.168

Concluding discussion



In this tutorial, we introduce conformal predictive systems as an extension of the Conformal Prediction toolbox in KNIME.

We have also shown how Mondrian conformal predictive systems can easily be achieved through a few additional nodes.

Having these strong tools easily accessible in a user-friendly platform such as KNIME increases the possibility of reaching user groups otherwise without access to the conformal framework.



JÖNKÖPING UNIVERSITY