Proceedings of Machine Learning Research 152:1–8, 2021 Conformal and Probabilistic Prediction and Applications

Conformal Prediction and its Integration within Visual Analytics Toolbox

Tomaž Hočevar TOMAZ.HOCEVAR@FRI.UNI-LJ.SI
Blaž Zupan BLAZ.ZUPAN@FRI.UNI-LJ.SI
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

Conformal prediction is a machine learning approach to report on the reliability of predictive models when applied to new cases. Machine learning techniques are gaining in complexity, and assessing their reliability may be an essential part of explaining the inner workings of predictive models. For practical purposes and dissemination of conformal prediction techniques, we must include these within easily accessible toolboxes. In machine learning, a significant subset of such toolboxes is those that use workflows and visual programming. Here, we report on an example of such a toolbox, Python implementation of conformal prediction library, and our initial efforts and ideas to democratize conformal prediction.

Keywords: conformal prediction, regression, visual programming, workflows, interactive analytics

1. Introduction

Machine learning and data mining are increasingly essential components of every researcher's toolbox. Most scientists and data owners have limited computer science skills and benefit from data exploration tools with rich graphical user interfaces. Such tools help to reduce the time to access advanced data analytics and increase transparency in understanding the data analytics process and the resulting model.

A particular set of tools aims to improve flexibility by providing components and means for the construction of workflows. Workflow construction is usually performed through visual programming, where a user selects the components and places them in a workflow to design a particular data analysis pipeline. Each component implements a specific data manipulation, visualization, or modeling task. Prominent examples of such frameworks include RapidMiner, KNIME, WEKA, and Orange, and they vary in depth of the toolboxes, openness of the software, and degree of interactivity of visualizations. Despite their differences, they aim to democratize data science, flatten the learning curve for novice users, and empower data owners with machine learning and techniques from data science.

We here briefly present a particular tool called Orange[1] (Demšar et al., 2013) and discuss and present its possible extension with techniques for conformal prediction. Conformal prediction Shafer and Vovk (2008) can determine precise levels of confidence in new predictions, and hence may be a precious tool for assessing reliability in predictive machine learning. Namely, in fields like chemoinformatics, precision medicine, and pharmaceutical

[1] https://orangedatamining.com/

sciences, for example, we can trust machine learning models if they are both accurate and report on the reliability of individual predictions. The conformal prediction would thus be a welcome addition to the visual programming data science environment. Their incorporation within the visual programming frameworks we have mentioned above could promote this interesting but under-used approach.

2. Orange – The Visual Programming Toolbox for Data Science

Orange is an open-source machine learning and data visualization software. The development of Orange began in 1997 at the Artificial Intelligence Laboratory and is currently taking place at the Laboratory of Bioinformatics, both at the University of Ljubljana (Demšar and Zupan, 2013). Orange focuses on implementing and including the most helpful and commonly used techniques. It focuses both on user-friendliness, flexibility, and interactive visualizations. The latest version, Orange3, is almost entirely implemented in Python and integrates NumPy and scikit-learn libraries for the core implementations of most data wrangling and machine learning. According to a recent poll, Orange is actively used by students, teachers, and researchers at more than five hundred higher education institutions from all over the world. Orange averaged about two thousand daily active users in July 2021. Hence, the integration of conformal prediction within Orange could also be seen as a vehicle to present this critical tool to a broader audience.

2.1. Extending Orange

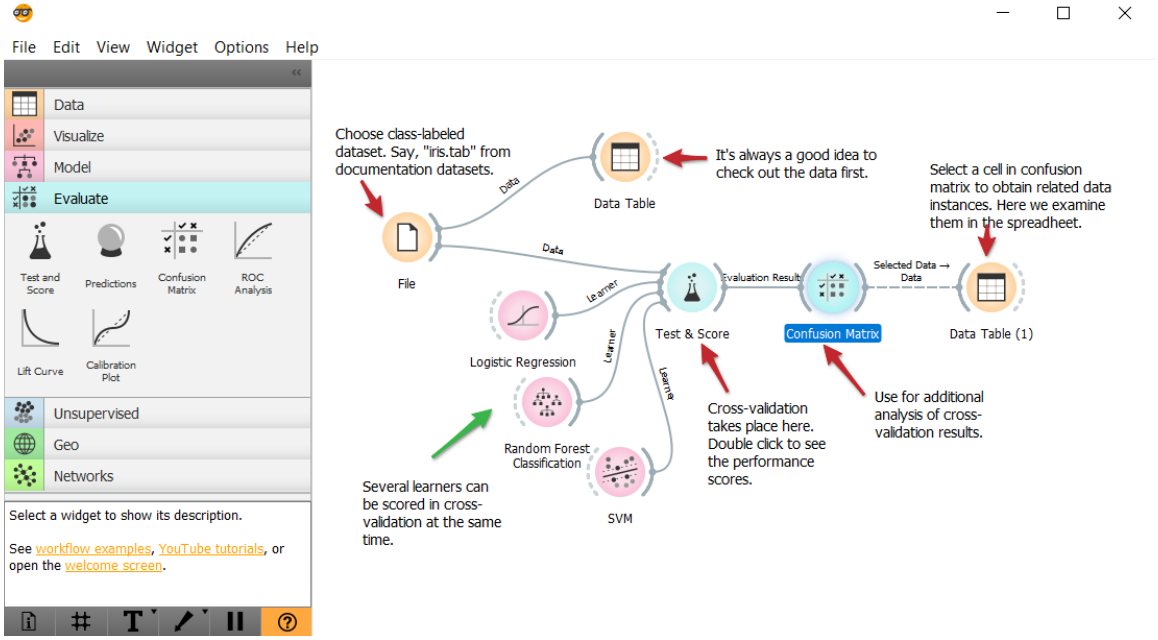
We designed Orange as an extensible package with add-ons that provide functionality for more specific use cases. Initially, the most prominent add-on was that for bioinformatics, with the tools that for analysis of gene expression data and inclusion of various biomedical databases and ontologies (Curk et al., 2005; Kokošar et al., 2020). Recent significant extensions, include those for single-cell gene expression analytics[2] Stražar et al. (2019), analysis of spectroscopy data[3] Toplak et al. (2017), image analytics (Godec et al., 2019), explainable machine learning, text mining, and other. Third-parties also develop prominent extensions. An example of such an extension is AZOrange, implementing techniques that were used in discovery, development and manufacturing of pharmaceutical drugs by Astra-Zeneca (Stålring et al., 2011).

2.2. Exploratory Data Analysis and Visual Programming

The emphasis of Orange is interactive data exploration. A simple visual programming interface and rich visualization methods support discovering exciting data patterns and enable rapid prototyping of machine learning models. Orange's user interface lets users focus on data analysis instead of dealing with the sometimes intricate details of programming. Orange users analyze data by constructing a data analysis workflow. Workflows are composed of components, also called widgets, and their connections, Figure 1. Each widget accepts an input of a given type and performs a task such as data transformation, model fitting, model evaluation, or visualization. The output of the widget is then passed on to other

[2] https://singlecell.biolab.si/

[3] https://quasar.codes/



Select a widget to show its description.

See workflow examples, YouTube tutorials, or open the welcome screen.

Choose class-labeled dataset. Say, "iris.tab" from documentation datasets.

It's always a good idea to check out the data first.

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadheet.

Cross-validation takes place here. Double click to see the performance scores.

Use for additional analysis of cross-validation results.

Several learners can be scored in cross-validation at the same time.

Figure 1: Screenshot of Orange with a number of connected widgets on the canvas.

downstream widgets. There are many ways of combining the widgets and building a complex data analysis pipeline. Each widget implements a user interface to set the parameters of the implemented method or observe the associated data visualization. Each change of parameters or change in the data selection in the interactive visualization automatically propagates to the downstream widgets of the workflow.

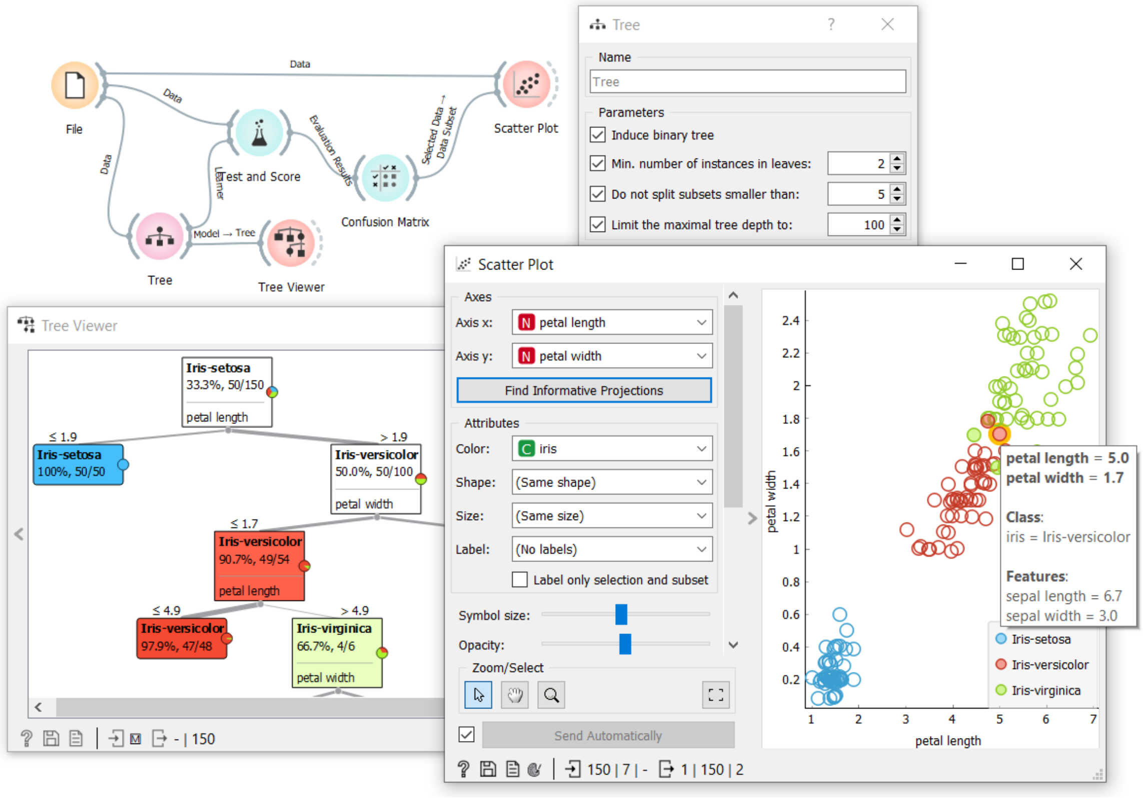
2.3. Interactive Visualizations

A picture is worth a thousand words; therefore, good visualization can provide more information than a table of numbers reporting the results. Orange offers a wide range of visualizations from scatter plots, box plots, histograms, and heat maps to model-specific visualizations such as classification tree viewers, hierarchical clustering dendrograms, and various embedding methods such as PCA or MDS. As data comes in different forms, add-ons visualize networks, word clouds, geographic data, and time series.

Most visualizations in Orange are interactive. Users can select parts of visualizations, for example, subsets of data instances or parts of data models. Orange then passes the data associated with the selection to downstream components for further analysis, inspection, or visualization. Figure 2, for example, shows a workflow where we train and test a tree classifier, inspect the tree model, select misclassified instances and visualize them (as filled circles) in a scatter plot.

2.4. Support for Training and Education

An important aspect of Orange is its convenience of the tool when teaching machine learning. Its interactive visualizations help understand the inner workings of machine learning models



Tree

Name: Tree

Parameters

☑ Induce binary tree

☑ Min. number of instances in leaves: 2

☑ Do not split subsets smaller than: 5

☑ Limit the maximal tree depth to: 100

Scatter Plot

Axes

Axis x: petal length

Axis y: petal width

Find Informative Projections

Attributes

Color: iris

Shape: (Same shape)

Size: (Same size)

Label: (No labels)

□ Label only selection and subset

Symbol size:

Opacity:

Zoom/Select

☑ Send Automatically

150 | 7 | - 1 | 150 | 2

petal length = 5.0
petal width = 1.7

Class:
iris = Iris-versicolor

Features:
sepal length = 6.7
sepal width = 3.0

Figure 2: Visualization of misclassified instances with a classification tree.

and, consequently, correct interpretation of the results. Orange allows us to illustrate rather than only explain. Additional widgets have been designed for this purpose and are available in the Orange3-Educational[4] add-on. A hands-on approach with a simple interface encourages experimentation and helps the students obtain practical experience. Orange has found a role in university courses on machine learning and data mining and for workshops aimed at industry and the public sector in a wide range of topics. In reasoning about conformal prediction in Orange, we consider the training aspect and design interfaces where conformal predictions are visualized, exposed, and potentially analyzed in the downstream workflow.

3. Conformal Prediction

Foundations of conformal prediction framework were set by Vovk et al. (2005) and popularized in a following tutorial (Shafer and Vovk, 2008). We implemented the basic concepts of transductive, inductive and cross-conformal predictors in combination with various nonconformity measures in the Orange3-Conformal (Hočevar et al., 2021) add-on[5] for the Orange3 package.

[4] https://pypi.org/project/Orange3-Educational/

[5] https://pypi.org/project/Orange3-Conformal/

Listing 1: Python Script

```
import numpy as np
from Orange.data import Table, Domain, DiscreteVariable
import orangecontrib.conformal as cp

np.random.seed(2021)

learn, predict = next(cp.evaluation.RandomSampler(in_data, 1, 1))
train, calibrate = next(cp.evaluation.RandomSampler(learn, 1, 1))
values = learn.domain.class_var.values

measure = cp.nonconformity.InverseProbability(in_learner)
predictor = cp.classification.InductiveClassifier(measure,
                                                  train, calibrate)

eps = 0.03
predictions = [predictor(instance, eps) for instance in predict]
results = [[value in prediction for value in values]
           for prediction in predictions]

new_attributes = tuple(DiscreteVariable(val, values=(”False”, ”True”))
                       for val in values)
domain = Domain(learn.domain.attributes + new_attributes,
                learn.domain.class_var)
out_data = Table.from_numpy(domain,
                            np.hstack((predict.X, results)), predict.Y)
```

The conformal prediction functionality is currently available only as a library for Python scripts and is not integrated with the graphical user interface, which is the most important part of Orange. However, we can use the Python Script widget, Listing 1, to achieve such integration. First, we import the necessary packages and set the random seed for replicability. Next, we split the data into a proper training set, a calibration set, and a prediction set, respectively. We also store the set of possible class values from the data set domain. To construct a conformal predictor, we use the inverse predicted probability from a provided learner as a nonconformity measure and use it in an inductive conformal classification setting. We choose a significance level and predict class values of every instance. These predictions are then transformed into True/False markers for every possible class value. Finally, the obtained results are appended to the data set as additional features and returned for further analysis.

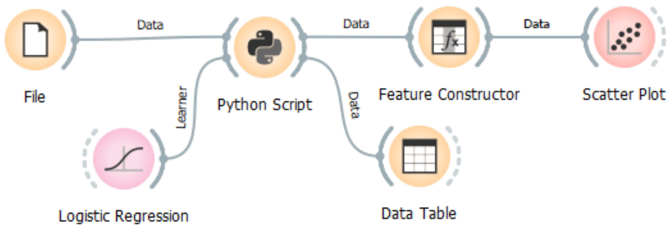
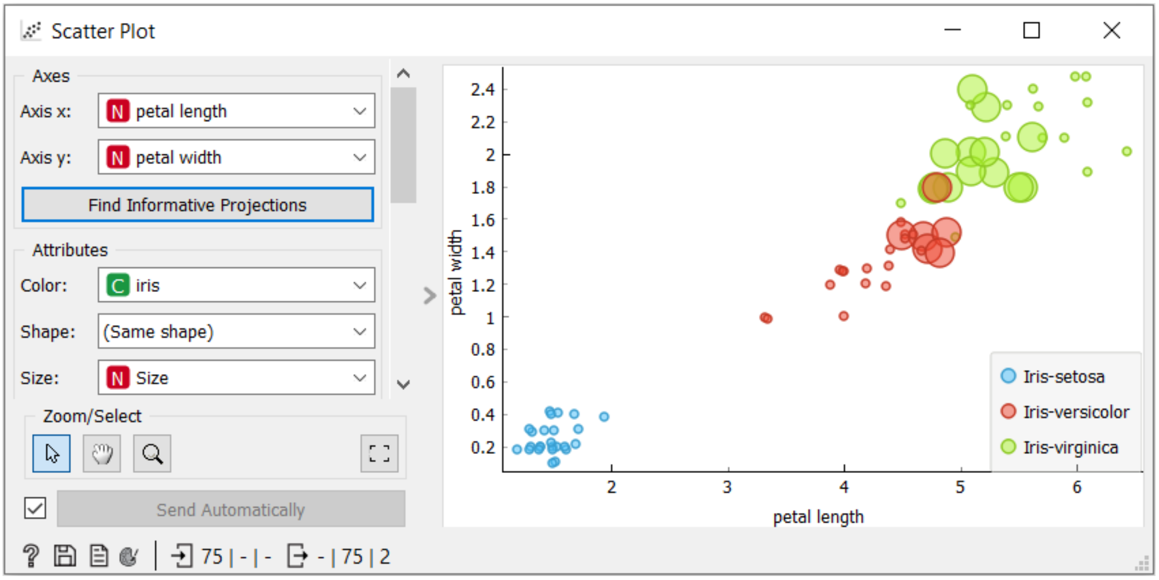


Figure 3: Orange workflow for conformal prediction

Such conformal prediction Python script returns the basic results in a data table that we can combine with other Orange widgets, Figure 3. For example, we can use a Data Table to inspect the data set or Feature Constructor to count the number of predicted class values. Data set, enhanced with the indicators of predicted class values, can be visualized in a Scatter Plot or other visualizations. Figure 4 shows that the instances with multiple predicted class values lie on the border between the `Iris-versicolor` and `Iris-virginica` classes.



Axis x: petal length

Axis y: petal width

Color: iris

Shape: (Same shape)

Size: Size

☑ Send Automatically

75 | - | -
- | 75 | 2

Figure 4: Scatterplot of instances, where the size of the dots indicates the number of predicted class values.

The above example illustrates a classification problem but can be easily modified for regression. Instead of producing several indicator columns for each class value, it would produce just two in regression, one for the lower and the other for the upper bound of the predicted interval.

4. Future Work

Integration of conformal prediction functionality with the Orange interface through the Python Script widget is the most basic way of including conformal prediction in our visual programming environment. While perfectly functional, ideally, the conformal prediction should be provided as a standalone widget. Existing Orange visualization widgets are general enough to provide insight into the workings and results of conformal prediction. However, we could further complement these visualizations with approaches explicitly designed for conformal prediction results. For example, many important methods are not yet implemented in the Orange3-Conformal package from conformal and related frameworks. Venn predictors (Vovk et al., 2005; Vovk and Petej, 2014) and conformal predictive distributions (Vovk et al., 2017) are two such examples. Implementation of these and designing Orange-related training material in combination with practical showcases for conformal prediction could constitute interesting future work, possibly within the scope of some interdisciplinary and collaborative project.

Acknowledgments

This work was supported by the Slovenian Research Agency (ARRS, grant number P2-0209) and by a research grant from Medivir AB.

References

Tomaž Curk, Janez Demšar, Qikai Xu, Gregor Leban, Uroš Petrovič, Ivan Bratko, Gad Shaulsky, and Blaž Zupan. Microarray data mining with visual programming. *Bioinformatics*, 21(3):396–398, 2005.

Janez Demšar and Blaž Zupan. Orange: Data mining fruitful and fun-a historical perspective. *Informatica*, 37(1), 2013.

Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, et al. Orange: data mining toolbox in python. *the Journal of machine Learning research*, 14(1):2349–2353, 2013.

Primož Godec, Matjaž Pančur, Nejc Ilenič, Andrej Čopar, Martin Stražar, Aleš Erjavec, Ajda Pretnar, Janez Demšar, Anže Starič, Marko Toplak, et al. Democratized image analytics by visual programming through integration of deep models and small-scale machine learning. *Nature communications*, 10(1):1–7, 2019.

Tomaž Hočevar, Blaž Zupan, and Jonna Stålring. Conformal prediction with orange. *Journal of Statistical Software*, 98(1):1–22, 2021.

Jaka Kokošar, Martin Stražar, Marko Toplak, Aleš Erjavec, and Lan Žagar. Visual programming and interactive visualisations for gene expression data analytics in orange. *Visual Programming and Interactive Visualisations for Gene Expression Data Analytics in Orange*, pages 121–134, 2020.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Jonna C Stålring, Lars A Carlsson, Pedro Almeida, and Scott Boyer. Azorange-high performance open source machine learning for qsar modeling in a graphical programming environment. *Journal of cheminformatics*, 3(1):1–10, 2011.

Martin Stražar, Lan Žagar, Jaka Kokošar, Vesna Tanko, Aleš Erjavec, Pavlin G Poličar, Anže Starič, Janez Demšar, Gad Shaulsky, Vilas Menon, et al. scorange—a tool for hands-on training of concepts from single-cell data analytics. *Bioinformatics*, 35(14): i4–i12, 2019.

M Toplak, G Birarda, S Read, C Sandt, SM Rosendahl, L Vaccari, J Demšar, and F Borondics. Infrared orange: connecting hyperspectral data with machine learning. *Synchrotron Radiation News*, 30(4):40–45, 2017.

Vladimir Vovk and Ivan Petej. Venn-abers predictors. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 829–838, 2014.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distributions based on conformal prediction. In *Conformal and Probabilistic Prediction and Applications*, pages 82–102. PMLR, 2017.