Calibrated Explanations for Multi-Class

Multi-class Explanations with Uncertainty

Tuwe Löfström, Helena Löfström, Ulf Johansson

09.09.2024

tuwe.lofstrom@ju.se
https://ju.se/personinfo.html?sign=loftuw
https://github.com/Moffran/calibrated_explanations

JÖNKÖPING UNIVERSITY
School of Engineering

Key Take-aways

Calibrated Explanations is a recently proposed feature importance explanation method providing uncertainty quantification

It utilises Venn-Abers to generate well-calibrated factual and alternative explanations for binary classification

In this paper, we extend the method to support multi-class classification using a recently proposed calibration strategy applicable to inherently multi-class models

The paper includes an evaluation illustrating the calibration quality of the selected multi-class calibration approach

The presentation focus on demonstrating how the explanations can help determine which explanations to trust

Uncertainty in Explanations

Explaining the process of machine learning models generating their predictions is crucial for increasing users' appropriate trust in high-stakes applications

A user who trusts the model too much accepts both correct and incorrect predictions with a risk of making low-quality decisions

Attaching confidence to the prediction and the contribution of its most relevant features can help users gain a better understanding of the model

- Increasing the possibilities for high-quality decisions

Probabilistic Prediction

Most predictive models producing probability estimates are often poorly calibrated

The standard procedure to handle poor calibration is to apply an external calibration method

The actual calibration is performed on a separate part of the available labelled data called the *calibration set*.

Calibration is when the following holds:

$$p(c_j \mid p^{c_j}) = p^{c_j},$$

where p^{c_j} is the probability estimate for class label c_j

Venn Predictors

Venn predictors are probabilistic predictors that output multiple probabilities for each label, with one of them being the valid one

In Venn prediction, instances are divided into a number of *categories*, based on a so-called *Venn taxonomy*

The relative frequency for each class label is calculated as the calibrated probability within the category of the test instance

To achieve validity, the test instance is also included in the calculation

Venn-Abers Predictors

- Finding a suitable Venn taxonomy is non-trivial

Venn-Abers prediction offers automated taxonomy optimisation using isotonic regression, providing optimised probability intervals for binary classification

One isotonic regressor is trained for each of the negative and the positive classes, defining the lower and upper bound of the multi-probability distribution

Since the instance must be one or the other, it follows that the interval must contain the true probability

Venn-Abers Predictors

One isotonic regressor is trained for each of the negative and the positive classes, defining the lower and upper bound of the multi-probability distribution

Since the instance must be one or the other, it follows that the interval must contain the true probability

A smaller interval indicates higher certainty about the prediction, while a larger interval indicates more uncertainty

Venn-Abers produces a lower and upper bound $[p_{low}, p_{high}]$ and a calibrated (regularised) probability estimate $p = \frac{p_{high}}{1 - p_{low} + p_{high}}$

Multi-class Calibration

The standard approach to multi-class problems has been to use either a one-vs-all or an all-vs-all schema

1. Requiring training multiple models
2. Calibrating each class
3. Aggregating the results into probability estimates

The possibility to analyse and explain a single model is no longer available

Multi-class Calibration

In a recently proposed method[1], multi-class calibration is performed on inherently multi-class models

The calibration is done by first calibrating each class label, in a one-vs-all calibration

The class label with the highest calibrated probability estimate is used as the predicted class

The multi-probabilistic prediction from Venn-Abers for the predicted class against all other classes is also returned

[1] Ulf Johansson, Tuwe Löfström, and Henrik Boström. Well-calibrated and sharp interpretable multi-class models. MDAI, 2021

Calibrated Explanations

Calibrated Explanations produce instance based explanations, explaining the factual prediction or exploring alternatives

The core of the algorithm is defined based on a numeric prediction and an uncertainty interval

For binary classification, the numeric prediction is a probability estimate of the positive class calibrated using a Venn-Abers calibrator, producing a lower and an upper bound for the calibrated probability estimate

Regression is supported using Conformal Predictive Systems, which is combined with Venn-Abers for thresholded probabilistic explanations ($y \leq t$)

Calibrated Explanations

A *factual explanation* is composed of:

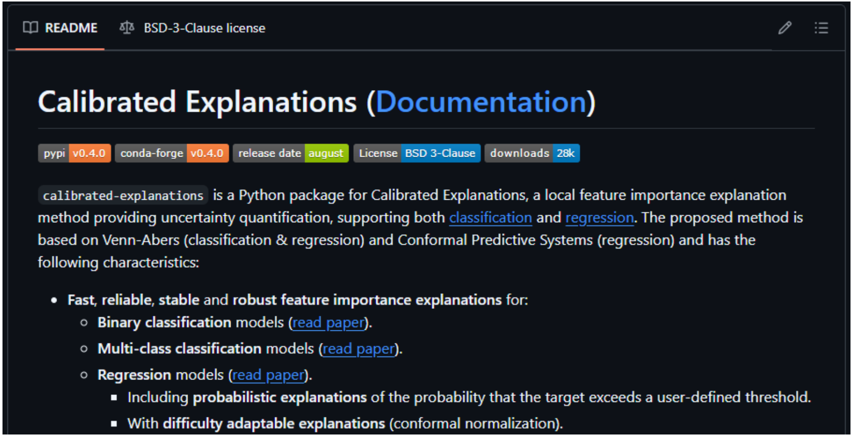
- A *calibrated prediction* from the underlying model accompanied by an *uncertainty interval*
- A collection of *factual feature rules*, each composed of a *feature weight with an uncertainty interval* and a *factual condition*, covering that feature's instance value

Alternative explanations[2] contain a collection of *alternative feature rules*, each composed of a *prediction estimate with an uncertainty interval* and a *alternative condition*, covering alternative instance values for the feature

[2] Referred to as *counterfactual explanations* in the current version

Calibrated Explanations

https://github.com/Moffran/calibrated_explanations



Calibrated Explanations (Documentation)

`calibrated-explanations` is a Python package for Calibrated Explanations, a local feature importance explanation method providing uncertainty quantification, supporting both classification and regression. The proposed method is based on Venn-Abers (classification & regression) and Conformal Predictive Systems (regression) and has the following characteristics:

- **Fast**, **reliable**, **stable** and **robust feature importance explanations** for:
  - **Binary classification** models (read paper).
  - **Multi-class classification** models (read paper).
  - **Regression** models (read paper).
    - Including **probabilistic explanations** of the probability that the target exceeds a user-defined threshold.
    - With **difficulty adaptable explanations** (conformal normalization).

Contribution: Calibrated Explanations for Multi-class

The contribution builds upon the multi-class calibration approach suggested in previous research

At initialisation a Venn-Abers calibrator is initialised for each class label c, using that label as the positive class and the probability estimate for that label as the score s^c from the underlying model

At prediction, identify the class with highest calibrated probability estimate, using the calibrators for each class

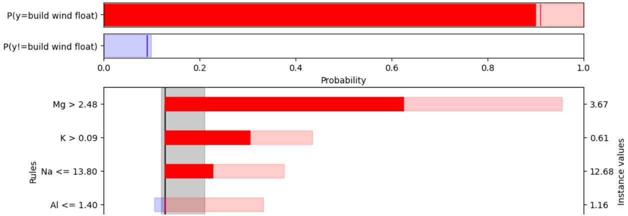
Use Calibrated Explanations with that class as positive class in a one-vs-all explanation

Calibrated Explanations for Multi-class

Code example on using *calibrated-explanations*. The same function calls are used for binary and multi-class classification as well as for regression

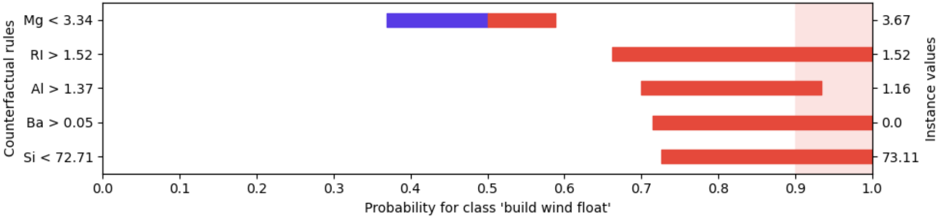
```python
from calibrated_explanations import WrapCalibratedExplainer
# Load and pre-process your data
# Divide it into proper training, calibration, and test sets

# Initialize the WrapCalibratedExplainer with your model
model = WrapCalibratedExplainer(ModelOfYourChoice())

# Train your model using the proper training set
model.fit(X_proper_training, y_proper_training)

# Calibrate your model using the calibration set
model.calibrate(X_calibration, y_calibration)

# Create and plot factual explanations
factual_explanations = model.explain_factual(X_test)
factual_explanations.plot()
factual_explanations.plot(uncertainty=True)

# Explore and plot alternative explanations
alternative_explanations = model.explain_counterfactual(X_test)
alternative_explanations.plot()
```

Factual explanation explained



- The top red (blue) bar shows the calibrated probability of the class predicted by Venn-Abers (any other class)
- The lower box contains the most important factual conditions with feature importance in bright colour and uncertainty in lighter colour
- Red feature importance favours the probability for the predicted class, blue feature importance is counter-indicative
- The actual feature value of the instance can be seen to the right of each bar
- The factual condition for which the feature importance applies can be seen to the left of the bar

Alternative Explanation explained



- The explanation shows five alternative conditions indicating what the probability interval for the class predicted by Venn-Abers would have been if the instance would have had a alternative feature value, as indicated by the condition
- The lighter coloured area in the background correspond to the prediction interval from Venn-Abers on the actual instance values
- The first alternative condition, `Mg < 3.34`, indicate that the belief in the predicted class would drop to approximately $[0.37, 0.59]$, i.e., become inconclusive

Method

The paper contains an evaluation of the calibration for multi-class method[3]

It shows the same general picture as in the introductory paper, skipped here

Focus is instead on a demonstration of the explanations and how they can be used[4]

[3] Code for running the experiments can be accessed at the GitHub repository for *calibrated-explanations* in the evaluation sub-folder

[4] All plots used for demonstrations can be found in `calibrated-explanations/notebooks/plots` and code for generating the plots in the notebook `demo_multiclass_glass.ipynb` in the notebooks folder

Confusion Matrix for the *glass* Data Set

Knowing the class distribution and which classes are easiest and which are hardest provides important cues for interpreting the explanations generated, as only the predicted class is shown in the explanation

		Predicted						
Full class names	**Actual**	**bwf**	**bwnf**	**c**	**h**	**t**	**vwf**	**Recall**
build wind float	**bwf**	**18**	4	0	0	0	0	0.818
build wind non-float	**bwnf**	4	**18**	0	1	1	0	0.750
containers	**c**	0	1	**2**	0	0	0	0.667
headlamps	**h**	1	0	0	**7**	0	0	0.875
tableware	**t**	0	0	0	0	**2**	0	1.000
vehic wind float	**vwf**	2	2	0	0	0	**0**	0.000
	Precision	0.720	0.720	1.000	0.875	0.667	NaN	

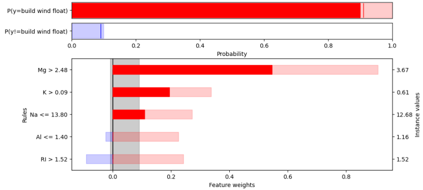
- *build wind float* (*bwf*) and *build wind non-float* (*bwnf*) are most common
- *tableware* (*t*) is entirely correctly predicted
- The least correct class is the *vehic wind float* (*vwf*), with none of the four instances being correctly predicted
- When looking at the precision, *vwf* has no predictions whatsoever, while both instances predicted as containers are in fact containers

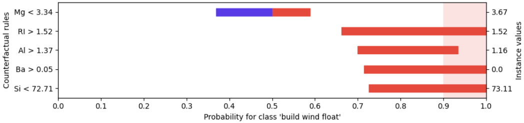
Explanation 1 for *build-wind-float*

Factual: The probability for the predicted class is very high, with the uncertainty interval indicated by the lighter red part

Alternatives: All alternative conditions provide at least fairly strong support for the predicted class

Interpretation: Combined with the evidence in the confusion matrix, it is easy to see that we have strong indication that the instance is correctly predicted



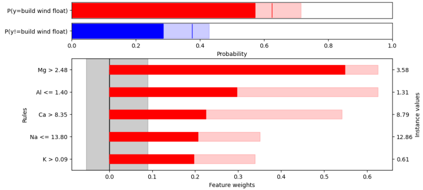


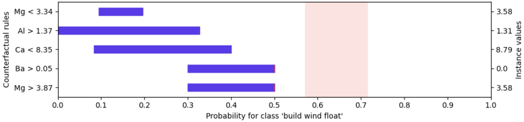
Explanation 2 for *build-wind-float*

Factual: The calibrated probability of the class predicted by Venn-Abers is much lower, with a wider uncertainty interval. All factual conditions and their feature importance and uncertainty interval indicate support for the positive class, sometimes with a large degree of uncertainty.

Alternatives: All the conditions substantially decrease the already low support for the predicted class

Interpretation: The explanations rather indicate against the prediction being accurate



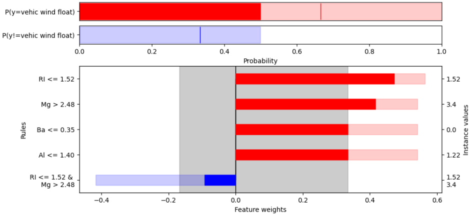


Explanation for *vehic-wind-float*

No calibration instances were predicted as *vwf*, making this explanation interesting

Factual: The prediction has strong support for the predicted class, even though the uncertainty is large[5]. The four first factual conditions clearly support the predicted class

Interpretation: Considering that this is a very difficult class and the explanation show strong support for the predicted class, it is reasonable to trust the prediction



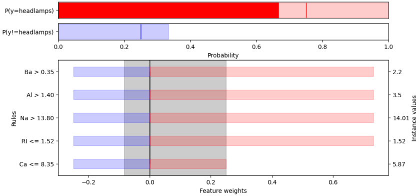
[5] Only four calibration instances belong to the *vwf* class

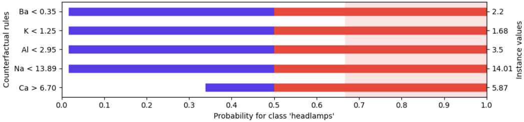
Explanation 1 for *headlamps*

Factual: Strong support for the predicted class with a lot of uncertainty. The first four factual conditions for instance 8 provide support for the predicted class *h*, even though the uncertainty is extremely large.

Alternatives: the first four alternative conditions all have 100% uncertainty, i.e., indicating less belief and more uncertainty in the predicted class *h*

Interpretation: Combining the evidence of the factual and the alternative explanations, the evidence in favour of the predicted class *h* is decreased



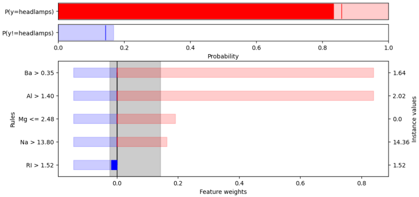


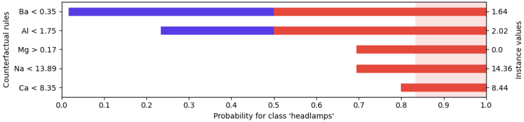
Explanation 2 for *headlamps*

Factual: Strong support for the positive class. Only two factual conditions provide strong support (with extreme uncertainty) for the predicted class *h*

Alternatives: The first two alternative conditions are very uncertain. The remaining alternative conditions for instance 14 are all favouring the predicted class *h*

Interpretation: The alternative explanation indicates that only two of the suggested changes would potentially change the support against the predicted class (even though the uncertainty is extreme), strengthening the belief in the prediction





Concluding Discussion

The paper provides a solution for how to provide calibrated explanations with uncertainty estimates for multi-class problems

The proposed solution combine previous work on multi-class calibration and calibrated explanations, providing a useful tool for analysis and decision support when working with multi-class problems

We have demonstrated how factual and alternative explanations could be used in combination with a confusion matrix to provide sufficient insights to enable a decision maker to determine which predictions to trust and which to distrust

The suggested approach is available as a python library installable using `pip` or `conda`. Code and examples can be found in the *evaluation* and *notebooks* folders

`https://github.com/Moffran/calibrated_explanations`

JÖNKÖPING UNIVERSITY