ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Model Agnostic Local Explanations of Reject

André Artelt*[†], Roel Visser and Barbara Hammer [‡]

CITEC – Cognitive Interaction Technology
Bielefeld University – Faculty of Technology
Inspiration 1, 33619 Bielefeld – Germany

Abstract. The application of machine learning based decision making systems in safety critical areas requires reliable high certainty predictions. Reject options are a common way of ensuring a sufficiently high certainty of predictions. While being able to reject uncertain samples is important, it is also of importance to be able to explain why a particular sample was rejected. However, explaining reject options is still an open problem. We propose a model-agnostic method for locally explaining reject options by means of interpretable models and counterfactual explanations.

1 Introduction

Nowadays, machine learning (ML) based decision making systems are omnipresent – in particular, they are used in safety critical scenarios such as autonomous driving [1] and credit (risk) assessment [2]. Trust and reliability are critical aspects of such decision making systems. Trust can be realized by transparency – i.e. it is difficult to trust a system that we do not understand. It is common to achieve transparency by means of explanations – i.e. providing explanations of the system's behavior [3]. Reliability means that we require the system to consistently output high quality predictions. However, because the models are build to output a prediction for every possible input, a high quality prediction cannot always be guaranteed. Uncertain predictions are problematic in scenarios where making mistakes can have serious consequences – in such cases it might be better to refuse to output a prediction instead of making a potentially wrong prediction [4]. For instance consider the example of a spam and phishing mail filter: *The filter is supposed to automatically sort out mails where it is certain that the particular mails are malicious. However, in cases where the filter is not absolutely certain about its classification, it should reject this mail and pass it to a human for manual inspection of its content – e.g. rejected mails might be passed to the user with an additional warning. In order to understand the rejection and to support the further development of the filtering application, it is helpful to get an explanation of why the filter was not able to classify the given mail.* To the best of our knowledge, the only existing work on explaining rejects is [5], which deals with reject options for learning vector quantization (LVQ) models. However, their proposed method is completely tailored towards LVQ models – i.e. it is not applicable to any other models or reject options. In this work, we propose a model-agnostic method for locally explaining any type of reject option.

*Corresponding author: aartelt@techfak.uni-bielefeld.de

[†] Affiliation with the University of Cyprus

[‡] We gratefully acknowledge fundings from the German Research Foundation for grant TRR 318/1 2021 - 438445824, and the VW-Foundation for the project *IMPACT*.

2 Foundations

Reject Options Given an arbitrary classifier $h : \mathcal{X} \to \mathcal{Y}$, a reject option [6] is usually added by providing an additional function $r_h : \mathcal{X} \to \mathbb{R}_+$ that measures the certainty of classifying $\vec{x}$ and rejects a sample $\vec{x}$ if the certainty is below a given threshold θ: $r_h(\vec{x}) < \theta$ .We construct a new classifier $h' : \mathcal{X} \to \mathcal{Y} \cup \{\emptyset\}$ where we add a reject symbol $\emptyset$ to the set of possible predictions $\mathcal{Y}$:

$$h'(\vec{x}) = \begin{cases} h(\vec{x}) & \text{if } r_h(\vec{x}) \geq \theta \\ \emptyset & \text{otherwise} \end{cases} \tag{1}$$

Conformal Prediction for Implementing a Reject Option Assume that a (black-box) probabilistic classifier $h : \mathcal{X} \to \mathcal{Y}$ of the following form is given: $h(\vec{x}) = \underset{y \in \mathcal{Y}}{\arg\max}\; p(y \mid \vec{x})$, where $p(y \mid \vec{x})$ denotes the class wise probability as estimated by the classifier $h(\cdot)$. A central building block of a conformal predictor [7] is a so called non-conformity measure $\phi_h : \mathcal{X}, \mathcal{Y} \to \mathbb{R}$ which measures how different a given labeled sample is from a given set of labeled samples we have seen before. In case of a probabilistic classifier $h(\cdot)$, a common non-conformity measure is given as follows: $\phi_h(\vec{x}, y = j) = \max_{i \neq j} p_h(y = i \mid \vec{x}) - p_h(y = j \mid \vec{x})$. For calibrating (i.e. fitting) a conformal predictor based on $h(\cdot)$, we need another labeled data set $\mathcal{D}_{\text{calib}} \subset \mathcal{X} \times \mathcal{Y}$ which was not used during the fitting of $h(\cdot)$ – we compute the non-conformity α_i of every sample from the calibration set by applying $\phi_h(\cdot)$: $\alpha_i = \phi_h(\vec{x}_i, y = y_i)$. For every new data point $\vec{x}_* \in \mathcal{X}$ that is going to be classified, we compute the non-conformity measure for every possible label in $\mathcal{Y}$. Next, the non-conformity scores of $\vec{x}_*$ are compared with the non-conformity scores from the calibration set to compute p-values $p_{y=i}(\cdot)$ for every possible classification of $\vec{x}_*$. The conformal predictor then selects the label with the largest p-value as a prediction: $h(\vec{x}_*) = \underset{i \in \mathcal{Y}}{\arg\max}\; p_{y=i}(\vec{x}_*)$. The credibility of the prediction – i.e. how well the training set supports the prediction – is given by the largest p-value: $\psi(\vec{x}_*) = \max_i p_{y=i}(\vec{x}_*)$. We implement a reject option Eq. (1) [4] using the credibility score:

$$r_h(\vec{x}) = \psi(\vec{x}) = \max_i p_{y=i}(\vec{x}) \tag{2}$$

Explanation Methodologies There exist popular methods for locally explaining a given model $h(\cdot)$, instead of trying to come up with a global explanation [3]. A common approach for local explanations is to build a local approximation of the model $h(\cdot)$ which is then used for creating an explanation. A popular instance of such methods is LIME [8], where an interpretable model is fit to a set of labeled perturbed samples – the labeling is done using the original model. The final local explanation is then constructed using the most relevant features of the local approximation – in order to get a meaningful explanation, the features must be interpretable and meaningful (e.g. super-pixels in case of images). Another method based on local approximations is Anchors [9], which compute if-then rules based explanations that locally explain the prediction of $h(\cdot)$.

Counterfactual explanations (often just called *counterfactuals*) [10] are a prominent instance of contrasting explanations, which state a change to some features of a given input such that the resulting data point, called the counterfactual, causes a different behavior of the system than the original input does. Thus, one can think of a counterfactual explanation as a suggestion of actions that change the model's behavior/prediction. One reason why counterfactual explanations are so popular is that there exists evidence that explanations used by humans are often contrasting in nature [11] – i.e. people often ask questions like *"What would have to be different in order to observe a different outcome?"*. In order to keep the explanation (suggested changes) simple – i.e. easy to understand – an obvious strategy is to look for a small number of changes so that the resulting sample $\vec{x}_{\text{cf}}$ (counterfactual) is similar/close to the original sample $\vec{x}_{\text{orig}}$. This is aimed to be captured by the following optimization problem [10]: $\underset{\vec{x}_{\text{cf}} \in \mathbb{R}^d}{\arg\min}\ \ell\left(h(\vec{x}_{\text{cf}}), y'\right) + C \cdot \theta(\vec{x}_{\text{cf}}, \vec{x}_{\text{orig}})$, where $\ell(\cdot)$ denotes a loss function, y' the target prediction, $\theta(\cdot)$ a penalty for dissimilarity of $\vec{x}_{\text{cf}}$ and $\vec{x}_{\text{orig}}$, and $C > 0$ denotes the regularization strength. In the following, we assume a binary classification problem: In this case, we denote a counterfactual $\vec{x}_{\text{cf}}$ of a given sample $\vec{x}_{\text{orig}}$ under a prediction function $h(\cdot)$ simply as $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}_{\text{orig}}, h)$ and drop the target label y' because it is uniquely determined.

3 Local Approximations for Explaining Reject

We propose a model-agnostic approach for locally explaining arbitrary reject options – i.e. our method does not need access to the reject option or the underlying ML model, access to a prediction interface is sufficient. Instead of explaining the reject option globally, we aim for a local explanation – i.e. explaining the reject of a particular sample. Given a sample $\vec{x}_{\text{orig}} \in \mathcal{X}$ which is rejected by the reject option, we sample a fixed number of samples $\{\vec{x}_i\}$ from the neighborhood around $\vec{x}_{\text{orig}}$ and label each sample whether it is also rejected or not:

$$y_i = \begin{cases} 1 & \text{if } r(\vec{x}_i) < \theta \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, \vec{x}_i \in \mathcal{B}_\epsilon(\vec{x}_{\text{orig}}) \tag{3}$$

where $\mathcal{B}_\epsilon(\vec{x}_{\text{orig}})$ denotes a fixed number of samples in the neighborhood of $\vec{x}_{\text{orig}}$. Then, we fit an interpretable classifier h_{local} (e.g. a linear model or a decision tree) to these samples $\mathcal{D}_{\text{local}} = \{(\vec{x}_i, y_i)\}$. We propose to either use $h_{\text{local}}(\cdot)$ as an explanation – e.g. using the obtained feature importances or learned decision rules as an explanation –, or a counterfactual explanation $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}_{\text{orig}}, h_{\text{local}})$ of $h_{\text{local}}(\cdot)$ as an explanation of the reject of $\vec{x}_{\text{orig}}$.

Formally, we propose two different realizations of a local explanation Ψ at $\vec{x}_{\text{orig}}$ under a given reject option $r(\cdot)$:

$$\Psi(r, \vec{x}_{\text{orig}}) = \begin{cases} \text{FRI}(h_{\text{local}}) \\ \text{CF}(\vec{x}_{\text{orig}}, h_{\text{local}}) \end{cases} \tag{4}$$

where $\text{FRI}(\cdot)$ denotes the feature relevance as obtained from a given model. We empirically evaluate and compare both types of explanations in Section 4.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

4 Experiments

We empirically evaluate our proposed explanations under two different aspects: computational aspects like sparsity and accuracy – i.e. checking if the original sample is also rejected under the local approximation – of the computed explanations; ground truth recovery rate (goodness) of the explanations by evaluating if and how well the explanations match the ground truth – i.e. identifying the relevant features. The implementation of the experiments is available on GitHub[1].

4.1 Data Sets

We consider the following data sets for our empirical evaluation – all data sets are scaled and standardized: "Wine data set" [12], "Breast cancer data set" [13] "Flip data set" [14], "t21 data set" [15].

4.2 Setup

Since our proposed explanation methodology is completely model-agnostic, we evaluate it on a set of diverse classifiers: k-nearest neighbors classifier (kNN), Gaussian naive Bayes classifier (GNB), random forest classifier (RandomForest). We always use conformal prediction for realizing a credibility based reject option Eq. (2). We perform hyperparameter tuning by a grid search and try to find an appropriate rejection threshold by using the Knee/Elbow method [16]. We run all experiments (combination of data sets and classifiers) in a 5-fold cross validation. We use a decision tree as an interpretable local approximation. After fitting the classifier, we apply the reject option to all samples from the test set and compute explanations for those that are rejected by the reject option. We always compute two explanations Eq. (4): feature relevance profile according to the Gini importance from the decision tree classifier, and a counterfactual under the local approximation.

When evaluating algorithmic properties, we not only compute the accuracy of the local approximation, but also compute the sparsity (l_0-norm) of both explanations. For evaluating the goodness of the explanations, we create scenarios with known ground truth as follows: For each data set, we select a random subset of features (30%) and perturb these in the test set by adding Gaussian noise – we then check which of these samples are rejected due to the noise and compute explanations of these samples only. Finally, we evaluate for both explanations how many of the relevant features are recovered and included in the explanation.

4.3 Results & Discussion

We use the following abbreviations: *FeatImp* – Feature importances as obtained from the local approximation; *Cf* – Counterfatual explanation.

Algorithmic Properties We report the mean accuracy and sparsity in Table 1. We observe that the local approximation is usually sufficiently good and the final explanations are very sparse – i.e. we obtain low-complexity explanations.

[1] `https://github.com/andreArtelt/LocalModelAgnosticExplanationReject`

Table 1: Algorithmic properties – Mean (incl. variance) accuracy and sparsity – larger values are better for accuracy, while smaller values are better for sparsity.

	DataSet	Accuracy	FeatImp	Cf
kNN	Wine	0.80 ± 0.16	4.5 ± 1.98	1.25 ± 0.23
	Breast Cancer	0.92 ± 0.00	5.12 ± 1.66	1.25 ± 0.19
	t21	0.96 ± 0.00	3.90 ± 3.43	1.07 ± 0.27
	Flip	0.31 ± 0.07	5.21 ± 1.13	1.00 ± 0.00
GNB	Wine	0.92 ± 0.00	4.57 ± 1.17	1.11 ± 0.10
	Breast Cancer	0.88 ± 0.00	3.83 ± 1.38	1.07 ± 0.07
	t21	0.78 ± 0.15	1.12 ± 1.71	0.71 ± 0.26
	Flip	0.83 ± 0.01	1.73 ± 0.54	1.00 ± 0.00
RandomForest	Wine	0.80 ± 0.16	3.26 ± 1.64	1.43 ± 0.37
	Breast Cancer	1.00 ± 0.00	1.07 ± 2.35	0.52 ± 0.48
	t21	0.95 ± 0.00	3.75 ± 2.59	1.22 ± 0.30
	Flip	0.50 ± 0.06	5.05 ± 1.33	1.05 ± 0.05

Table 2: Goodness of explanations – Mean (incl. variance) recall of correctly identified relevant features (larger numbers are better).

	DataSet	Accuracy	FeatImp	Cf
kNN	Wine	0.75 ± 0.15	0.53 ± 0.03	0.28 ± 0.15
	Breast Cancer	0.89 ± 0.02	0.50 ± 0.04	0.23 ± 0.12
	t21	0.78 ± 0.02	0.56 ± 0.03	0.36 ± 0.15
	Flip	0.40 ± 0.14	0.30 ± 0.08	0.04 ± 0.03
GNB	Wine	0.85 ± 0.04	0.56 ± 0.06	0.43 ± 0.18
	Breast Cancer	0.97 ± 0.00	0.39 ± 0.09	0.23 ± 0.12
	t21	0.60 ± 0.24	0.45 ± 0.13	0.36 ± 0.15
	Flip	0.91 ± 0.01	0.40 ± 0.14	0.38 ± 0.18
RandomForest	Wine	1.00 ± 0.00	0.51 ± 0.13	0.39 ± 0.16
	Breast Cancer	1.00 ± 0.00	0.18 ± 0.08	0.16 ± 0.09
	t21	0.62 ± 0.11	0.58 ± 0.05	0.50 ± 0.15
	Flip	0.61 ± 0.08	0.54 ± 0.08	0.38 ± 0.15

Furthermore, we observe that counterfactual explanations of the local approximation are consistently sparser than the obtained feature importance.

Goodness of Explanations The mean recall of correctly recovered relevant features is given in Table 2. We observe that the perturbation does not strongly affect the accuracy. However, both explanations have trouble to recover all perturbed features – although the feature importance explanation recovers consistently more perturbed features than the counterfactual explanation, which is due to the sparsity objective. In addition, it seems that the local approximation is not sensitive enough to the applied perturbations – the accuracy is pretty high, but still the explanations have trouble identifying all perturbed features.

5 Summary & Conclusion

In this work, we proposed a model-agnostic approach for explaining reject options, by using local interpretable approximation of the reject option and explain the reject locally either by the local approximation itself or by counterfactuals of this local approximation. We empirically evaluated these two explanations under computational as well as qualitative aspects. We observed reasonable performance of both explanations – in particular counterfactuals were able to come up with low complexity explanations but identified fewer of the relevant features. The empirical evaluation in this work focuses on computational proxies only. However, it still remains unclear if and how useful our proposed explanations are to humans. Since it is difficult to phrase "usefullness" as a scoring function, a proper use study is needed. We leave these aspects as future work.

References

[1] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19):70–76, 2017.

[2] Amir E. Khandani, Adlar J. Kim, and Andrew Lo. Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2010.

[3] Christoph Molnar. Interpretable Machine Learning. 2019.

[4] Henrik Linusson, Ulf Johansson, Henrik Boström, and Tuve Löfström. Classification with reject option using conformal prediction. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 94–105. Springer, 2018.

[5] André Artelt, Johannes Brinkrolf, Roel Visser, and Barbara Hammer. Explaining reject options of learning vector quantization classifiers, 2022.

[6] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. arXiv preprint arXiv:2107.11277, 2021.

[7] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. J. Mach. Learn. Res., 9:371–421, 2008.

[8] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In SIGKDD-2016, pages 1135–1144. ACM, 2016.

[9] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In AAAI-2018, pages 1527–1535. AAAI Press, 2018.

[10] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech., 31:841, 2017.

[11] Ruth M. J. Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In IJCAI-19, 2019.

[12] D. Coomans S. Aeberhard and O. de Vel. Comparison of classifiers in high dimensional settings. Tech. Rep. no. 92-02, 1992.

[13] Olvi L. Mangasarian William H. Wolberg, W. Nick Street. Breast cancer wisconsin (diagnostic) data set, 1995.

[14] Jan-Peter Sowa, Dominik Heider, Lars Peter Bechmann, Guido Gerken, Daniel Hoffmann, and Ali Canbay. Novel algorithm for non-invasive assessment of fibrosis in nafld. PLOS ONE, 8(4):1–6, 04 2013.

[15] K. H. Nicolaides, K. Spencer, K. Avgidou, S. Faiola, and O. Falcon. Multicenter study of first-trimester screening for trisomy 21 in 75 821 pregnancies: results and estimation of the potential impact of individual risk-orientated two-stage first-trimester screening. Ultrasound in Obstetrics & Gynecology, 25(3):221–226, 2005.

[16] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In 2011 31st International Conference on Distributed Computing Systems Workshops, pages 166–171, 2011.