Improving Anchor-based Explanations

Julien Delaunay
Inria/IRISA
France
julien.delaunay@irisa.fr

Luis Galárraga
Inria/IRISA
France
luis.galarraga@inria.fr

Christine Largouët
Agrocampus Ouest/IRISA
France
christine.largouet@irisa.fr

ABSTRACT

Rule-based explanations are a popular method to understand the rationale behind the answers of complex machine learning (ML) classifiers. Recent approaches, such as Anchors, focus on local explanations based on if-then rules that are applicable in the vicinity of a target instance. This has proved effective at producing faithful explanations, yet anchor-based explanations are not free of limitations. These include long overly specific rules as well as explanations of low fidelity. This work presents two simple methods that can mitigate such issues on tabular and textual data. The first approach proposes a careful selection of the discretization method for numerical attributes in tabular datasets. The second one applies the notion of pertinent negatives to explanations on textual data. Our experimental evaluation shows the positive impact of our contributions on the quality of anchor-based explanations.

KEYWORDS

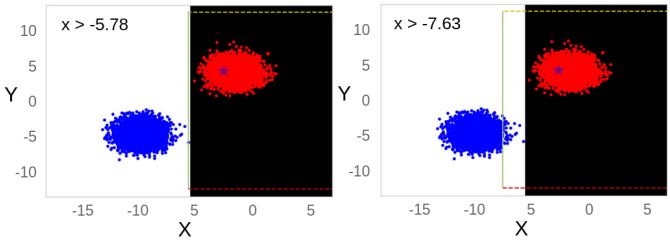
Explainable AI, Machine Learning, Rule Mining, Discretization

ACM Reference Format:
Julien Delaunay, Luis Galárraga, and Christine Largouët. 2018. Improving Anchor-based Explanations. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Explanations based on logical rules are a popular strategy to explain the logic of complex black-box machine learning (ML) classifiers. However, approximating a complex model with human-readable rules incurs an inevitable trade-off: Fidelity can only be achieved at the expense of complexity, and complex explanations miss the whole point of explainable ML. For this reason recent approaches, such as Anchors [Ribeiro et al. 2018], focus on explanations of local scope. These are if-then rules – also called *anchors* – that mimic the black box in the vicinity of a target instance. This strategy relies on the assumption that the black-box classifier is simpler to approximate when we focus on a particular region of the space.

While local rule-based explanations yield and simple and locally faithful explanations, their quality can still be very sensitive to some design factors. One of such factors is the discretization of the numerical attributes for tabular data. Figure 1 illustrates the anchors obtained for the same dataset with two discretization methods. When running Anchors with a suitable discretization method on the left-hand side of the figure, we obtain the anchor $x > -5.78 \Rightarrow Red$ that matches the black box's behavior more faithfully than the anchor obtained by another discretization method on the right-hand side.



(a) A good discretization **(b) A suboptimal discretization**

Figure 1: Two anchors (depicted as green lines) learned with different discretizations of the numerical features. The target instance is marked as a violet star

Another factor that can impact the quality of an anchor is the training set used to learn the explanation. Anchors [Ribeiro et al. 2018] generates training samples by perturbing the instance of interest according to a neighborhood generation strategy. Figure 2 shows the average anchor length (number of conditions on the rule's antecedent) and precision across 10 instances of three explanations learned with different neighborhood generation methods. The strategy in dark blue (*pertinent negatives* explained later) provides the explanation with the best trade-off between rule length and precision.

In this work we study the impact of discretization and neighborhood generation on the different metrics that define the quality of anchor-based explanations. Our contributions focus on the tabular and text variants of Anchors and include (i) the application of MDLP [Fayyad and Irani 1993] to discretize the numerical attributes on tabular data, and (ii) the definition of *pertinent negatives* on text classifiers. Before elaborating on our contributions, we provide a proper introduction to Anchors in the next section.

2 ANCHORS

Consider a black-box classifier $f : \mathcal{X}^d \rightarrow \mathcal{Y}$ that maps instances $x \in \mathcal{X}^d$ to a set $\mathcal{Y}$ of classes. Each instance x is a vector of d attributes that can be either categorical or numerical, and $x[j]$ denotes the value of x for the j-th attribute. [Ribeiro et al. 2018] defines an *anchor* as a logical rule R that explains a black-box f

Authors' addresses: Julien Delaunay, Inria/IRISA, France, julien.delaunay@irisa.fr; Luis Galárraga, Inria/IRISA, France, luis.galarraga@inria.fr; Christine Largouët, Agrocampus Ouest/IRISA, France, christine.largouet@irisa.fr.

0730-0301/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

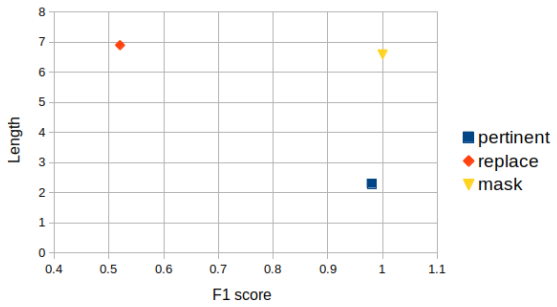


Figure 2: Trade-off between F1 score and anchor length for three neighborhood generation methods

around a target instance x. The anchors operates on a surrogate space defined via a conversion function $\eta : \mathcal{X}^d \rightarrow \{0,1\}^{d'}$. The generated rule has the form:

$$R : \mathbf{B} \Rightarrow f(\eta^{-1}(z)) = f(x) \qquad \text{with } \mathbf{B} = \bigwedge_{j \in F \subseteq \{1,\ldots,d'\}} z[j]$$

The left-hand side (or antecedent) of the rule is a conjunction of conditions that predicts $f(x)$, i.e., the class of the target instance x according to the black box. An example is the rule $age \in [28, 37] \wedge$ *workclass*="*private*" $\Rightarrow$ "*well-paid*" (for simplicity we write only the predicted value on the right-hand side). For tabular data, the interpretable space can be obtained by discretizing the numerical variables – to turn them categorical – and then binarizing the resulting conditions. For text classifiers, the surrogate space is usually defined on the presence or absence of words.

The method proposed by [Ribeiro et al. 2018], learns rule-based explanations from the answers of the black box f on a randomly generated neighborhood $\mathcal{Z} \subseteq \{0,1\}^{d'}$ constructed around $z = \eta(x) \in \mathcal{Z}$. Anchors applies principles of depth-first search and multi-armed bandit theory to output the shortest anchor with the largest coverage that satisfies the precision guarantee $prec(R) = P(f(\eta^{-1}(z)) = f(x) \mid \mathbf{B} \wedge z \in \mathcal{Z}) \geq \tau$ for a user-defined precision threshold τ. The coverage of an anchor is the ratio of instances in $\mathcal{Z}$ that match the anchor's antecedent, i.e., $cov(R) = P(\mathbf{B} \mid z \in \mathcal{Z})$.

The next two sections present our contributions that highlight some of the limitations of Anchors and propose improvements.

3 IMPACT OF DISCRETIZATION ON TABULAR DATA

The variant of Anchors for tabular data assumes we have access to the black box's training dataset $\mathcal{D} \subseteq \mathcal{X}^d$. Tabular Anchors uses $\mathcal{D}$ to discretize the numerical attributes properly according to the data distribution. It supports three discretization methods. Two of them, *decile* and *quartile*, are based on classical quantile discretization. In contrast, the *entropy* discretization method splits the domain of an attribute j in a dataset $\mathcal{D}$ (denoted by $\mathcal{D}[j]$) so that the information entropy of $f(x)$ – for $x \in \mathcal{D}$ and black box f – is minimized. Anchors' entropy-based discretization outperforms quantile-based discretization in terms of coverage, precision, and anchor length. However, as we show later in this section, it can still lead to relatively long anchors. On these grounds we investigate the performance of two new discretization methods.

3.1 New Discretization Methods

3.1.1 K-means. We propose a baseline discretization method based on the k-means clustering algorithm [Jain and Dubes 1988]. This method splits the domain $\mathcal{D}[j]$ of an attribute into k clusters that minimize intra-cluster distance while maximizing inter-cluster distance. Distance is based on the absolute difference of the values in $\mathcal{D}[j]$. Therefore, and unlike the entropy-based method, our adaptation of k-means does not make use of the labels provided by the black box f. The parameter k is chosen using the Elbow method [Thorndike 1953].

3.1.2 MDLP Discretization. [Fayyad and Irani 1993] proposes a method for discretization of continuous-valued attributes into multiple intervals based on the Minimum Description Length Principle (MDLP). Intuitively, MDLP returns the minimal number of "pure" intervals needed to separate instances from distinct classes. Compared to a traditional entropy-based method, MDLP focuses on compression minimality, hence it outputs as few intervals as possible. Its key heuristic lies in the selection of the "best" cut points.

3.2 Experimental Evaluation

We evaluate the quality of Anchors when used with five discretization methods. This includes the three methods already supported, i.e., quartile (Q), decile (D), entropy (E), and the methods proposed in this work, i.e., MDLP (M) and k-means (K). The precision threshold τ (Section 2) is set to the default value, that is, $\tau = 0.95$.

3.2.1 Metrics. The quality of an anchor is defined by its *coverage*, *precision*, and *length*. Shorter anchors with high coverage and precision are preferred. We highlight that the coverage of an anchor is almost analogous to the *recall*: It defines a trade-off with the precision. The only difference with respect to recall is that coverage is not necessarily bounded by 1. To account for this issue, we define the normalized coverage $ncov(R)$ of an anchor R as follows:

$$ncov(R) = \frac{cov(R)}{P(f(\eta^{-1}(z)) = f(x) \mid z \in \mathcal{Z})}$$

That is, the standard coverage is now normalized by the maximal attainable coverage of an anchor that explains the class given by $f(x)$. With this formulation we can define the F1 measure of an anchor as the harmonic mean of the precision and the normalized coverage. This score provides a trade-off between coverage and precision.

3.2.2 Datasets. We use three synthetic and two real datasets for our evaluation. The synthetic datasets were generated by drawing 10k instances with the functions `make_blobs`, `make_moons`, and `make_circles` available in scikit-learn[1]. The real datasets comprise (i) *Titanic*[2], where the goal is to predict if a passenger of the Titanic survived based on her age, sex, class, etc., and (ii) *Adult*[3] where we aim at predicting if a person earns more than 50k USD also based on personal characteristics.

[1] https://scikit-learn.org
[2] https://www.kaggle.com/c/titanic/data
[3] https://archive.ics.uci.edu/ml/datasets/adult

	Support Vector Machines					Logistic Regression					Multilayer Perceptron					Random Forest				
	K	M	D	Q	E	K	M	D	Q	E	K	M	D	Q	E	K	M	D	Q	E
Blobs	0.89	**1**	0.84	0.85	**1**	0.89	**1**	0.84	0.85	**1**	0.89	**1**	0.84	0.85	**1**	0.89	**1**	0.84	0.85	**1**
Circles	0.45	**0.87**	0.43	0.46	0.77	0.45	**0.87**	0.43	0.46	0.77	0.45	**0.87**	0.43	0.46	0.77	0.45	**0.87**	0.43	0.46	0.77
Moons	0.6	0.7	0.66	**0.72**	0.68	0.6	0.7	0.66	**0.72**	0.68	0.6	0.7	0.66	**0.72**	0.68	0.6	0.7	0.66	**0.72**	0.68
Adult	0.66	0.66	0.66	**0.98**	0.66	0.66	0.96	0.66	**0.98**	0.66	0.66	0.96	0.66	**0.98**	0.66	0.66	0.65	0.66	**0.98**	0.66
Titanic	0.42	**0.93**	0.33	0.42	0.42	0.42	**0.93**	0.33	0.42	0.42	0.42	**0.93**	0.33	0.42	0.42	0.42	**0.93**	0.33	0.42	0.42
MR	3	**1.4**	3.6	2	2	3.2	**1.4**	3.8	2	2.2	3.2	**1.4**	3.8	2	2.2	3	**1.6**	3.6	2	2

Table 1: F1 score for Anchors using different discretization methods. MR denotes the mean rank of the method.

	Support Vector Machines					Logistic Regression					Multilayer Perceptron					Random Forest				
	K	M	D	Q	E	K	M	D	Q	E	K	M	D	Q	E	K	M	D	Q	E
Blobs	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Circles	8.14	**2.39**	4.79	3.69	3.69	8.14	**2.39**	4.79	3.69	3.69	8.14	**2.39**	4.79	3.69	3.69	8.14	**2.39**	4.79	3.69	3.69
Moons	2.47	2.46	**1.95**	2.49	2.72	2.47	2.46	**1.95**	2.49	2.72	2.47	2.46	**1.95**	2.49	2.72	2.47	2.46	**1.95**	2.49	2.72
Adult	9.37	7.43	7.6	**6.48**	8.54	9.16	8.03	7.89	**6.41**	8.23	8.99	7.26	8.34	**6.67**	8.49	9.11	7.21	8.26	**6.77**	8.84
Titanic	4.18	**2.58**	4.72	3.52	3.52	4.18	**2.58**	4.72	3.52	3.52	4.18	**2.58**	4.72	3.52	3.52	4.18	**2.58**	4.72	3.52	3.52
MR	3.2	**1.4**	2.4	2	2.8	3.2	**1.6**	2.2	2	2.4	3.2	**1.4**	2.4	2	2.8	3.2	**1.4**	2.4	2	2.8

Table 2: Anchor length using different discretization methods. MR denotes the mean rank of the method.

3.2.3 Black-box models. We tested our contributions on a variety of black-box classifiers, namely logistic regression, support vector machines, multi-layer perceptron, and random forests.

3.3 Results

Table 1 summarizes the F1 performance of Anchors for a set of instances[4] of each dataset for all the studied black-box models and discretization methods. The labels K, M, D, Q, and E denote k-means, MDLP, decile, quartile, and entropy respectively. We observe that MDLP achieves overall the best F1, followed by quartile and entropy. In particular, MDLP and quartile split the domain of attributes into fewer intervals, leading to less specific conditions with potentially higher coverage. The use of the black-box labels for binning gives MDLP a significant advantage over a simple quartile discretization. Besides, the focus on compression minimality makes MDLP output fewer intervals than the *entropy* strategy. Table 2 confirms our intuitions as we observe that MDLP yields on average the shortest anchors. All discretization methods obtain a good performance on the highly structured dataset *Blobs* (depicted in Figure 1). An example of an anchor using MDLP in the *Adult* dataset is $age \leq 22 \wedge relationship = \text{"own-child"} \Rightarrow < 50kUSD$.

4 IMPROVING THE QUALITY OF ANCHORS ON TEXT

In some of our experiments with Anchors on text data, it was impossible to attain the default precision threshold $\tau = 0.95$. This phenomenon makes Anchors output rules with a precision smaller than τ. We argue that the maximal attainable precision of Anchors depends on (i) the distribution of the training neighborhood and the expressiveness of the rule language. In this section we study the performance of Anchors for two different neighborhood generation strategies and propose an extension of the rule language by considering negated conditions, known as pertinent negatives in the explainable AI literature.

[4] 10k for the synthetic datasets, 100 for Titanic and Adult.

4.1 Neighborhood Generation Strategies

The variant of Anchors for text classification converts a textual instance into a surrogate binary vector where each entry defines the absence or presence of a word of the target phrase. Consider, for example, a black-box classifier f for sentiment analysis and the target instance "This is a good book". Anchors will convert this instance into a five-component vector, i.e., 11111, and generate neighbors by randomly toggling off bits of this binary representation. Examples are the instances 10101 or 11101. An anchor is induced from that set of neighbors and their class labels according to the black box f. However, f operates in a different space than Anchors. Hence, the inverse conversion function η^{-1} must map the generated neighbors to actual text instances. The strategy called *mask words* does so by replacing the words of each zero component with a neutral wildcard unseen before by the black box. In our example the neighbor 11101 becomes "This is a $\mathbb{W}$ book" for wildcard $\mathbb{W}$. The strategy called *replace words*, on the other hand, replaces toggled-off words with random words that have the same syntactic role, i.e., they would be assigned the same part-of-speech tag. For instance the neighbor 11101 could become "This is a **great** book".

4.2 Pertinent Negatives

We highlight that anchors are defined on conjunctions of non-negated conditions. For text data this entails conditions on the presence of words in phrases. This design decision guarantees simpler rules while keeping the search space under control. On the downside, it imposes limits on the expressiveness of explanations. Inspired by the work presented by [Dhurandhar et al. 2018], we propose to change the language of Anchors and provide explanations on the absence of words. Those words are known as *pertinent negatives*. We can also see pertinent negatives as counterfactual explanations or words whose presence would change the answer of the black box.

Considering the absence of all possible words in the corpus makes the search space for anchors prohibitively large. Hence we apply two mechanisms to alleviate this fact. First, we focus on a limited set of words. This set consists of the top k most frequent

	Support Vector Machines			Logistic Regression			Multlayer Perceptron			Random Forest		
	RW	MW	PN	RW	MW	PN	RW	MW	PN	RW	MW	PN
Tweets	5.1	**4.3**	8.1	**4.6**	6.4	9.1	**2.1**	4.6	7.2	**3.7**	5.4	4.2
Polarity	8.3	7.7	**4.9**	6.7	**3.6**	4.5	6.9	6.6	**2.3**	**2**	**2**	**2**

Table 3: Length of textual Anchors for different neighborhood generation strategies.

	Support Vector Machines			Logistic Regression			Multilayer Perceptron			Random Forest		
	RW	MW	PN	RW	MW	PN	RW	MW	PN	RW	MW	PN
Tweets	0.63	0.41	**0.82**	0.56	0.31	**0.91**	0.85	0.44	**0.95**	0.57	0.27	**0.95**
Polarity	0.35	**1**	0.87	0.35	**0.98**	0.86	0.52	**1**	0.98	0.47	**1**	0.79

Table 4: F1 score of textual Anchors for different neighborhood generation strategies.

words that co-occur next to the words of the target instance, stopwords excluded. For our example phrase "This is a good book", our algorithm would consider words such as *scientific*, *interesting*, or *very* as they may often appear with "book" and "good". Second, we set an upper bound n in the number of pertinent negatives allowed in explanations.

It follows that a neighborhood generation method purely based on pertinent negatives represents a phrase as a vector of $m+p$ components where m is the number of words in the target phrase and p is the number of pertinent negatives. The target instance is mapped to a vector where the first m elements are set to 1 and the remaining are set to 0. Neighbors are then generated by randomly toggling on the zero entries of the pertinent negatives, which instructs Anchors to add the word to the phrase. Our goal is to show the potential and viability of pertinent negatives in Anchors, thus we leave as future work the implementation of a hybrid approach that combines pertinent negatives with one of the classical strategies for neighborhood generation based on present words.

4.3 Experimental Evaluation

We evaluate the discussed neighborhood generation strategies using the F1 measure and the anchor length as quality criteria. For pertinent negatives we use $n = 20$.

4.3.1 Datasets. Our experimental datasets comprise (i) *Polarity*[5], a set of movie reviews for sentiment analysis, and (ii) *Tweets*[6], a set of tweets, where the goal is to predict the occurrence of emojis.

4.3.2 Black-box models. We use the same black-box models as in Section 3.2.3. Those models were trained on a vector representation of the phrases based on word counts and provided by the class `CountVectorizer` of scikit-learn.

4.4 Results

We summarize the aggregated results for the F1 measure and the anchor size among 10 randomly selected instances in Tables 3 and 4. We first observe that the *replace words* strategy lies far behind the others in terms of F1 (except for logistic regression on *Tweets*). While it usually produces anchors of high precision, the coverage of those anchors is very low, in other words, it generates overly specific explanations. This intuition is confirmed by Figure 3, where we can observe that *replace words* yields, on average, longer anchors than the other strategies. These overly specific anchors are a consequence of the neighborhood generation strategy. By replacing toggled-off words with other words of the same syntactic role, the neighbor instances become very unstable: the addition of a single word can change the meaning of the phrase as well as the black box's answer. We observe this phenomenon to a lesser extent when using the strategy *mask words*. This happens because replacing a word with the wildcard forces the black box to make a decision on the basis of fewer words. We observe that pertinent negatives achieves the best F1 score and leads to shorter and more stable anchors. This happens because the training set is based on the variation of a small and carefully selected set of features (words) that exhibit a high correlation with the features present in the target phrase. An example of an anchor with pertinent negatives is $\neg caring \wedge downpur \Rightarrow$ 😂 for the tweet "Totally worth getting caught in this evening's downpour. #jacquelineonassisreservoir". The addition of the word "caring" would have made the classifier predict a different emoji.

[5]http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6]https://competitions.codalab.org/competitions/17344#learn_the_details-data

5 CONCLUSION

In this work we have studied the impact of two elements in the design of anchor-based explanations, namely the discretization of numerical features for tabular data, and the neighborhood generation strategy for text data. We have shown that by careful adjusting those elements, we can obtain a significant increase in the precision, coverage, and length of anchor-based explanations. As future work, we envision to explore post-hoc discretization methods, i.e., methods that rediscretize the variables of an anchor in order to improve its coverage. We will also consider the combination of the pertinent negatives and mask words strategies in order to provide more expressive and accurate anchors defined both on the presence or absence of words. The code, data, and experimental results are available at https://github.com/juliendelaunay35000/anchors.

REFERENCES

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. 2018. Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. In *Annual Conference on Neural Information Processing Systems (NIPS 2018)*. 590–601.

Usama M. Fayyad and Keki B. Irani. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-1993)*. 1022–1029.

Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice-Hall.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*. 1527–1535.

Robert L. Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (01 Dec 1953), 267–276.