Anchors: High-Precision Model-Agnostic Explanations

Marco Tulio Ribeiro
University of Washington
marcotcr@cs.washington.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

Abstract

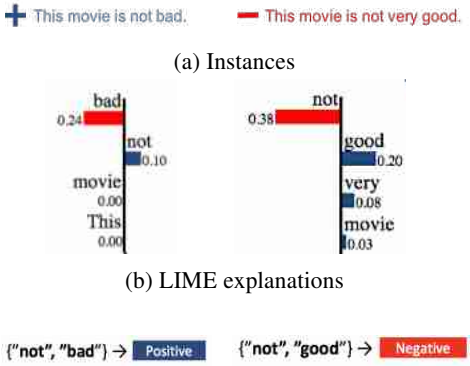
We introduce a novel model-agnostic system that explains the behavior of complex models with high-precision rules called *anchors*, representing local, "sufficient" conditions for predictions. We propose an algorithm to efficiently compute these explanations for any black-box model with high-probability guarantees. We demonstrate the flexibility of anchors by explaining a myriad of different models for different domains and tasks. In a user study, we show that anchors enable users to predict how a model would behave on unseen instances with less effort and higher precision, as compared to existing linear explanations or no explanations.

Introduction

Sophisticated machine learning models such as deep neural networks have been shown to be highly accurate for many applications, even though their complexity virtually makes them black-boxes. As a consequence of the need for users to understand the behavior of these models, *interpretable machine learning* has seen a resurgence in recent years, ranging from the design of novel *globally*-interpretable machine learning models (Lakkaraju, Bach, and Leskovec 2016; Ustun and Rudin 2015; Wang and Rudin 2015) to local explanations (for individual predictions) that can be computed for any classifier (Baehrens et al. 2010; Ribeiro, Singh, and Guestrin 2016b; Strumbelj and Kononenko 2010).

A question at the core of interpretability is whether humans understand a model enough to make accurate predictions about its behavior on unseen instances. For instances where humans can confidently predict the behavior of a model, let (human) *precision* be the fraction in which they are correct (note that this is human precision, not model precision). High human precision is paramount for real interpretability - one can hardly say they understand a model if they consistently think they know what it will do, but are often mistaken.

Most local approaches provide explanations that describe the local behavior of the model using a linearly weighted combination of the input features (Baehrens et al. 2010; Ribeiro, Singh, and Guestrin 2016b; Strumbelj and Kononenko 2010). Linear functions can capture relative importance of features in an easy-to-understand manner. However, since these linear explanations are in some way local, it is not clear whether they *apply* to an unseen instance. In other words, their coverage (region where explanation applies) is unclear. Unclear coverage can lead to low human precision, as users may think an insight from an explanation applies to unseen instances even when it does not. When combined with the arithmetic involved in computing the contribution of the features in linear explanations, the human effort required can be quite high.



(a) Instances

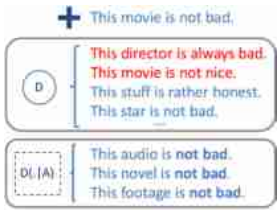
(b) LIME explanations

(c) Anchor explanations

Figure 1: Sentiment predictions, LSTM

Take for example LIME (Ribeiro, Singh, and Guestrin 2016b) explanations for two sentiment predictions made by an LSTM in Figure 1. Although both explanations are computed to be *locally accurate*, if one took the explanation on the left and tried to apply it to the sentence on the right, one might be tempted to think that the word "not" would have a positive influence, which it does not. While such explanations provide insight into the model, their coverage is not clear, e.g. when does "not" have a positive influence on sentiment?

In this paper, we introduce novel model-agnostic explanations based on if-then rules, which we call *anchors*. An *anchor* explanation is a rule that sufficiently "anchors" the prediction locally – such that changes to the rest of the feature values of the instance do not matter. In other words, for instances on which the anchor holds, the prediction is (almost) always the same. For example, the anchors in Figure 1c state that the presence of the words "not bad" virtually guarantee a prediction of positive sentiment (and "not good" of negative



(a) $\mathcal{D}$ and $\mathcal{D}(.|A)$

(b) Two toy visualizations

Figure 2: Concrete example of $\mathcal{D}$ in (a) and intuition (b)

Instance	If	Predict
I want to play(V) ball.	previous word is PARTICLE	play is VERB.
I went to a play(N) yesterday.	previous word is DETERMINER	play is NOUN.
I play(V) ball on Mondays.	previous word is PRONOUN	play is VERB.

Table 1: Anchors for Part-of-Speech tag for the word "play"

sentiment). Anchors are intuitive, easy to comprehend, and have extremely clear coverage – they only apply when all the conditions in the rule are met, and if they apply the precision is high (by design).

We demonstrate the usefulness of anchors by applying them to a variety of machine learning tasks (classification, structured prediction, text generation) on a diverse set of domains (tabular, text, and images). We also run a user study, where we observe that anchors enable users to predict how a model would behave on unseen instances with much less effort and higher precision as compared to existing techniques for model-agnostic explanation, or no explanations.

Anchors as High-Precision Explanations

Given a black box model $f : X \to Y$ and an instance $x \in X$, the goal of local model-agnostic interpretability (Ribeiro, Singh, and Guestrin 2016a; 2016b; Strumbelj and Kononenko 2010) is to explain the behavior of $f(x)$ to a user, where $f(x)$ is the individual prediction for instance x. The assumption is that while the model is globally too complex to be explained succinctly, "zooming in" on individual predictions makes the explanation task feasible. Most model-agnostic methods work by perturbing the instance x according to some "perturbation distribution" $\mathcal{D}_x$ (for simplicity, $\mathcal{D}$ from now on). In Ribeiro, Singh, and Guestrin (2016b), we emphasize that the perturbations $\mathcal{D}$ (and explanations) must use an interpretable representation (i.e. one that makes sense to humans), even if the model uses an alternative representation of the input.

Let A be a rule (set of predicates) acting on such an interpretable representation, such that $A(x)$ returns 1 if all its feature predicates are true for instance x. For example, in Figure 2a (top), $x =$ "This movie is not bad.", $f(x) =$ *Positive*, $A(x) = 1$ where $A = \{$"not", "bad"$\}$. Let $\mathcal{D}(\cdot|A)$ denote the conditional distribution when the rule A applies (e.g. similar texts where "not" and "bad" are present, Figure 2a bottom). A is an *anchor* if $A(x) = 1$ and A is a sufficient condition for $f(x)$ with high probability — in our running example, if a sample z from $\mathcal{D}(z|A)$ is likely predicted as *Positive* (i.e. $f(x) = f(z)$). Formally A is an anchor if,

$$\mathbb{E}_{\mathcal{D}(z|A)}[\mathbb{1}_{f(x)=f(z)}] \geq \tau, A(x) = 1. \quad (1)$$

Figure 2b shows two "zoomed in" regions of a complex model, with particular instances (+ and -) being explained. LIME (Ribeiro, Singh, and Guestrin 2016b) explanations work by learning the lines that best approximate the model under $\mathcal{D}$, with some local weighting. The resulting explanations give no indication of how faithful they are (the explanation on the right is a much better local approximation of the black box model than the one on the left), or what their "local region" is. In contrast, even though they use the same $\mathcal{D}$, anchors are by construction faithful, adapting their coverage to the model's behavior (the anchor on the right of Figure 2b is broader) and making their boundaries clear.

Leaving the discussion of how to compute anchors for later, we now demonstrate their usefulness and flexibility via concrete examples in a variety of domains and models.

Text Classification: We have already alluded to Figure 1, where we trained an LSTM model with paraphrastic sentence embeddings (Wieting et al. 2015) to predict the sentiment of reviews. In this case, the *features* used by the model are uninterpretable. The interpretable representation we use is the presence of individual tokens (words) in the instance. The perturbation distribution $\mathcal{D}$ replaces "absent" tokens by random words with the same POS tag with probability proportional to their similarity in an embedding space (Pennington, Socher, and Manning 2014) (i.e. the generated sentences are coherent, and of the same length). We show samples from $\mathcal{D}$ for a sentence in Figure 2a. The anchor $A = \{$"not", "bad"$\}$ is easy to apply: if the words "not" and "bad" are in the sentence, the model will predict "positive" with probability at least τ (set to 0.95 from here onwards). If either word (or both) are not present, we know we do not have sufficient information to know what the model will do. Examples from $\mathcal{D}(z|A)$ are shown in Figure 2a (bottom).

Structured prediction: When the output of the algorithm is a structure, the anchor approach can be used to explain any function of the output. Anchors are particularly suited for structured prediction models: while the global behavior is too complex to be captured by simple interpretable models, the local behavior can usually be represented using short rules.

In Table 1, we explain the predictions of a state of the art part-of-speech tagger for the word "play" in different contexts, where we include the tags for neighboring words as part of the interpretable representation. The anchors demonstrate that the model is picking up on reasonable patterns of the English language, e.g. if play is preceded by a determiner, it is likely used as a noun.

In Table 2, we compute anchors for a multi-layer RNN encoder/attention-based decoder translation system (Vinyals et al. 2015) trained on English-Portuguese parallel corpora. In this case, anchors explain the presence of a word (or certain words) in the translation. The anchors (in bold) are computed

English	Portuguese
This is the **question** we must address	Esta é a questão que temos que enfrentar
This is the **problem** we must address	Este é o problema que temos que enfrentar
This is what we must address	É isso que temos de enfrentar

Table 2: Anchors (in bold) of a machine translation system for the Portuguese word for "This" (in pink).

with respect to the presence of the words in pink in the Portuguese text. The first row in Table 2 means that when the words "This", "is", and "question" appear in English, the translation will include the word "Esta". In Portuguese, the translation for the word "this" depends on the gender of the word it refers to ("esta" for feminine, "este" for masculine), or should be "isso" if its referent is not in the sentence. The anchors show that the model is capturing this behavior as it always includes "this is", and the word that "this" refers to ("question" is feminine, "problem" is masculine).

Tabular Classification: Classification of data in tabular form (categorical and/or continuous features) is a popular application of machine learning. We use a validation dataset to define $\mathcal{D}$, and sample from $\mathcal{D}(z|A)$ by fixing the predicates in A and sampling the rest of the row as a whole. We show anchors for a few predictions of 400 gradient boosted trees trained on balanced versions of three datasets in Table 3. The anchors provide valuable insight into these models. Marital status appears in many different anchors for predicting whether a person makes $> \$50$K annually (*adult* dataset). The anchors for predicting recidivism for individuals released from prison (Schmidt and Witte 1988) (*rcdv* dataset), show that this model is unfair if used for bail decisions, as race and gender feature prominently. For predicting whether a loan on the Lending Club website will turn out bad, i.e. late payment or default (*lending* dataset), the FICO score is sufficient in the extremes, but loan amount is taken into consideration otherwise. We note that these are not exhaustive – the models are complex, and these anchors explain their behavior on part of the input space but not all of it. Anchors for "hard" predictions (in particular boundary cases) may be longer.

Image Classification: When explaining the label prediction for an image we follow Ribeiro, Singh, and Guestrin (2016b), segmenting the image into superpixels (Vedaldi and Soatto 2008) and using the presence or absence of these superpixels as the interpretable representation. In contrast to Ribeiro, Singh, and Guestrin (2016b), instead of *hiding* superpixels, we define $\mathcal{D}(z|A)$ by fixing the superpixels in A to the original image and superimposing another image over the rest of the superpixels. We explain a prediction of the Inception-V3 neural network (Szegedy et al. 2015) in Figure 3b. Even though $\mathcal{D}$ is quite unrealistic here, the anchor demonstrates that the model focuses on various parts of the dog to determine its breed. An inspection of the (admittedly bizarre) images from $\mathcal{D}(z|A)$ in Figure 3c for which the model predicts "beagle" with high confidence illustrates that attributes

	If	Predict
adult	No capital gain or loss, never married	≤ 50K
	Country is US, married, work hours > 45	> 50K
rcdv	No priors, no prison violations and crime not against property	Not rearrested
	Male, black, 1 to 5 priors, not married, and crime not against property	Re-arrested
lending	FICO score ≤ 649	Bad Loan
	$649 \leq$ FICO score ≤ 699 and $\$5,400 \leq$ loan amount $\leq \$10,000$	Good Loan

Table 3: Generated anchors for Tabular datasets

that humans would consider essential for a beagle prediction (legs, not being underwater, not being in the sky, not having a human body) are not quite as essential to the neural network.

Visual Question Answering (VQA): As a final example, we present anchors for the VQA task: answering a question asked of a reference image. Here, we are interested in identifying which part of the *question* led to the predicted answer, and thus use the same representation and distribution as in Figure 2b. We explain predictions from the Visual7W open-ended VQA system (Zhu et al. 2016) using Figure 3a as the reference image. In Figure 3d we show a question/prediction pair and its anchor (in bold), as well as samples from $\mathcal{D}(z|A)$ and their predictions. The short anchor ("What") reveals that, for this image, the model's answer to many questions will be "dog", contrary to our expectations. In Figure 3e, we show other question/answer pairs and their anchors, providing examples where the behavior of the classifier is aligned with our intuitions (first three) and where it is not (last question).

Related Work

Even in the few cases where having some understanding of a machine learning model's behavior is not a requirement, it is certainly an advantage. Relying only on validation accuracy has many well studied problems, as practitioners consistently overestimate their model's accuracy (Patel et al. 2008), propagate feedback loops (Sculley et al. 2015), or fail to notice data leaks (Kaufman, Rosset, and Perlich 2011).

Compared to other interpretable options, rules fare well; users prefer, trust and understand rules better than alternatives (Lim, Dey, and Avrahami 2009; Stumpf et al. 2007), in particular rules similar to anchors. Short, disjoint rules are easier to interpret than hierarchies like decision lists or trees (Lakkaraju, Bach, and Leskovec 2016). A number of approaches construct globally interpretable models, many based on rules (Lakkaraju, Bach, and Leskovec 2016; Letham et al. 2015; Wang and Rudin 2015; Wang et al. 2015). With such models, the user should be able to guess the model's behavior on any example (i.e. perfect coverage). However, these models are not appropriate for many domains, e.g. almost no interpretable rule-based system is suitable for text or image applications, due to the sheer size of the feature space, or are just not accurate enough. Interpretability, in

(a) Original image (b) Anchor for "beagle" (c) Images where Inception predicts $P(\text{beagle}) > 90\%$

What animal is featured in this picture ?	**dog**
What floor is featured in this picture?	dog
What toenail is paired in this flowchart ?	dog
What animal is shown on this depiction ?	dog

(d) **VQA:** Anchor (bold) and samples from $\mathcal{D}(z|A)$

Where is the **dog**?	on the floor
What color is the **wall?**	white
When was this picture taken?	during the day
Why is he lifting his paw?	to play

(e) **VQA:** More example anchors (in bold)

Figure 3: Anchor Explanations for Image Classification and Visual Question Answering (VQA)

these cases, comes at the cost of flexibility, accuracy, or efficiency (Ribeiro, Singh, and Guestrin 2016a). An alternative is learning a simple (interpretable) model to imitate the black box model globally (e.g. a decision tree (Craven and Shavlik 1996) or a set of rules (Sanchez et al. 2015)), but this may yield low human precision. Simple models are not able to fully capture the behavior of the complex ones, and thus lead users to wrong conclusions, especially since it is not clear when the simple model is faithful.

To avoid this, *local* model-agnostic explanations explain individual predictions (instead of the whole model at once). These methods provide a trade-off: each explanation is easy to understand even for complex models and tasks, but only captures the behavior of the model on a local region of the input space. The anchor approach falls within this category (and thus can explain complex models like translation and VQA), together with many forms of linear explanations (Baehrens et al. 2010; Ribeiro, Singh, and Guestrin 2016b; Strumbelj and Kononenko 2010). As illustrated pictorially by Figure 2b and concretely in Figure 1b, even the local behavior of a model may be extremely non-linear, leading to poor linear approximations and users being potentially misled as to how the model will behave (e.g. thinking that "not" will usually be positive after seeing Figure 1b (left)). Linear explanations are also harder to parse and apply than simple rules, as they involve mental calculations. Finally, even if the black-box model is approximately linear locally, humans may not be able to compute distance functions "correctly", and may think that a local explanation applies when it does not – the "unclear coverage" problem.

Anchors, on the other hand, make their coverage very clear — the user knows exactly when the explanation for an instance "generalizes" to other instances. By construction, anchors are not only faithful to the original model, but communicate its behavior to the user in such a way that virtually guarantees correct understanding, and thus high precision. Anchors are also able to capture non-linear behavior, even locally. In sum, the anchor approach combines the benefits of local model-agnostic explanations with the interpretability of rules, constructed in a way to best support human understanding.

Efficiently Computing Anchors

We revisit the problem definition from Eq. (1): given a black-box classifier f, instance x, distribution $\mathcal{D}$, and the desired level of precision τ, an anchor A is a set of feature predicates on x that achieves $\text{prec}(A) \geq \tau$, where

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)}\left[\mathbb{1}_{f(x)=f(z)}\right]. \tag{2}$$

For an arbitrary $\mathcal{D}$ and black-box model f, it is intractable to compute this precision directly. Instead, we introduce a probabilistic definition: anchors satisfy the precision constraint with high probability.

$$P\left(\text{prec}(A) \geq \tau\right) \geq 1 - \delta \tag{3}$$

If multiple anchors meet this criterion, those that describe the behavior of a larger part of the input space are preferred, i.e. ones with the largest *coverage* . Formally, we define the coverage of an anchor as the probability that it applies to samples from $\mathcal{D}$, i.e. $\text{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)]$.

We thus define this search for an anchor as the following combinatorial optimization problem:

$$\max_{A \text{ s.t. } P(\text{prec}(A)\geq\tau)\geq 1-\delta} \text{cov}(A). \tag{4}$$

The number of all possible anchors is exponential, and it is intractable to solve this problem exactly. While the search for anchors is similar in spirit to Probabilistic Inductive Logic Programming (ILP) (De Raedt and Kersting 2008) and other rule-finding methods, one crucial difference is that we do

not assume a dataset apriori - instead we have perturbation distributions and a black box model, which we can call to estimate precision and coverage bounds under $\mathcal{D}$. While we could in theory generate a very large dataset and then use methods like ILP to find anchors, the number of perturbed samples and predictions from the black box model would be prohibitive, especially in high-dimensional sparse domains such as text. In order to efficiently explore the model's behavior in the perturbation space, we turn to a multi-armed bandit formulation.

Bottom-up Construction of Anchors

We first introduce a bottom-up construction of anchors, which we later extend to search over a space of potential anchors. Here, we incrementally construct the anchor A, which is initialized with an *empty* rule, i.e. one that applies to every instance. In each iteration, we generate a number of candidate rules that extend A by one additional feature predicate, $\{a_i\}$, in its definition, i.e. the set of candidate rules in each iteration is $\mathcal{A} = \{A \wedge a_i, A \wedge a_{i+1}, A \wedge a_{i+2}, \ldots\}$. We identify the candidate rule with the highest *estimated precision* (as described next), replace A with the selected candidate, and repeat. If the current candidate rule meets the anchor definition in Eq. (3), we have identified our desired anchor and terminate. Although this approach does not directly compute the coverage, and instead tries to find the *shortest* anchor, we note that short anchors are likely to have a higher coverage, and require less effort from the users to understand.

In order to select the best candidate rule in each iteration, we want to estimate the precision of these candidates efficiently. Since we cannot compute the *true* precision, we rely on samples from $\mathcal{D}(\cdot|A)$ to estimate the precision of A; however, a fixed number of samples may be too many or too few for an accurate estimation. Instead, we are interested in a *minimal* set of calls to f (fewest samples from $\mathcal{D}$) in order to estimate which candidate rule has the highest *true* precision.

This problem can be formulated as an instance of *pure-exploration multi-armed bandit* problem (Kaufmann and Kalyanakrishnan 2013), i.e. each candidate A is an arm, the true precision of A on $\mathcal{D}(\cdot|A)$ is the *latent* reward, and each pull of the arm A is an evaluation of $\mathbb{1}_{f(x)=f(z)}$ on a sample from $\mathcal{D}(z|A)$. For such a setting, the KL-LUCB (Kaufmann and Kalyanakrishnan 2013) algorithm can be used to identify the rule with the highest precision. The algorithm works by constructing confidence regions based on KL divergence (Cover and Thomas 1991). In each step, the algorithm selects two distinct rules: the best mean (A) and the highest upper bound (A'), and updates their bounds by getting a sample each from $\mathcal{D}(z|A)$ and $\mathcal{D}(z'|A')$, and computing $\mathbb{1}_{f(x)=f(z)}$ and $\mathbb{1}_{f(x)=f(z')}$. This sampling process continues until the lower bound on A is higher than A''s upper bound with tolerance $\epsilon \in [0, 1]$. If A^* is the arm with highest true precision, the following (proved by Kaufmann and Kalyanakrishnan) holds for the true precision of the chosen rule A:

$$P(\text{prec}(A) \geq \text{prec}(A^*) - \epsilon) \geq 1 - \delta \tag{5}$$

Alg 1 presents an outline of this approach. When evaluating if the rule chosen by KL-LUCB is an anchor, we need to be confident it meets our precision criteria. Thus, if for an identified rule A, $\text{prec}_{lb}(A) < \tau$ but $\text{prec}_{ub}(A) > \tau$, we sample from $\mathcal{D}(\cdot|A)$ until either we are confident A is an anchor ($\text{prec}_{lb}(A) > \tau$) or not ($\text{prec}_{ub}(A) < \tau$).

Algorithm 1 Identifying the *Best* Candidate for Greedy

function GenerateCands($\mathcal{A}, c$)
  $\mathcal{A}_r = \emptyset$
  for all $A \in \mathcal{A}$; $a_i \in x, a_i \notin A$ **do**
    if $\text{cov}(A \wedge a_i) > c$ **then** {Only high-coverage}
      $\mathcal{A}_r \leftarrow \mathcal{A}_r \cup (A \wedge a_i)$ {Add as potential anchor}
  return $\mathcal{A}_r$ {Candidate anchors for next round}

function BestCand($\mathcal{A}, \mathcal{D}, \epsilon, \delta$)
  initialize $\text{prec}, \text{prec}_{ub}, \text{prec}_{lb}$ estimates $\forall A \in \mathcal{A}$
  $A \leftarrow \arg\max_A \text{prec}(A)$
  $A' \leftarrow \arg\max_{A' \neq A} \text{prec}_{ub}(A', \delta)$ {δ implicit below}
  while $\text{prec}_{ub}(A') - \text{prec}_{lb}(A) > \epsilon$ **do**
    sample $z \sim \mathcal{D}(z|A), z' \sim \mathcal{D}(z'|A')$ {Sample more}
    update $\text{prec}, \text{prec}_{ub}, \text{prec}_{lb}$ for A and A'
    $A \leftarrow \arg\max_A \text{prec}(A)$
    $A' \leftarrow \arg\max_{A' \neq A} \text{prec}_{ub}(A')$
  return A

Algorithm 2 Outline of the Beam Search

function BeamSearch($f, x, \mathcal{D}, \tau$)
  hyperparameters B, ϵ, δ
  $A^* \leftarrow$ **null**, $\mathcal{A}_0 \leftarrow \emptyset$ {Set of candidate rules}
  loop
    $\mathcal{A}_t \leftarrow \text{GenerateCands}(\mathcal{A}_{t-1}, \text{cov}(A^*))$
    $\mathcal{A}_t \leftarrow \text{B-BestCand}(\mathcal{A}_t, \mathcal{D}, B, \delta, \epsilon)$ {LUCB}
    if $\mathcal{A}_t = \emptyset$ **then break loop**
    for all $A \in \mathcal{A}_t$ s.t. $\text{prec}_{lb}(A, \delta) > \tau$ **do**
      if $\text{cov}(A) > \text{cov}(A^*)$ **then** $A^* \leftarrow A$
  return A^*

Beam-Search for Anchor Construction

Although the greedy approach we have described so far can find short anchors with the guarantee that, for each step, the choice was near optimal with high probability, it has two major shortcomings. First, due to the greedy nature of the approach, it is only able to maintain a single rule at a time (that it incrementally augments), and thus any suboptimal choice is irreversible. Second, the greedy algorithm is not directly concerned with the coverage of the anchors, and instead returns the shortest anchor that it finds. In order to address both these concerns, we extend the greedy approach to perform a beam-search by maintaining a set of candidate rules, while guiding the search to identify amongst many possible anchors the one that has the highest coverage.

The algorithm is outlined in Algorithm 2. It is similar in structure to the greedy approach, with a set of B current candidates instead of a single one. After generating all the possible candidates rules, we select the B-best candidates to keep based on the KL-LUCB approach with multiple arms (the *Explore-m* setting). For the tolerance $\epsilon \in [0, 1]$, this version of KL-LUCB algorithm returns a set $\mathcal{A}$ of size B that

is an ϵ-approximation of $\mathcal{A}^*$, with high probability.

$$P(\min_{A \in \mathcal{A}} \text{prec}(A) \geq \min_{A' \in \mathcal{A}^*} \text{prec}(A') - \epsilon) \geq 1 - \delta \quad (6)$$

We omit the description of KL-LUCB for this setting, but the intuition is similar to the one in the greedy approach. Further, amongst multiple anchors that we encounter, we output the one with the highest coverage, thus directly optimizing Eq. (4). This condition is also used for efficient pruning of the search space – we do not store any rule that has a lower coverage than that of the best anchor found so far, since the coverage of a rule can only reduce as more predicates are added. The beam-search algorithm is therefore more likely to return an anchor with a higher coverage than the one found by the greedy approach, and thus we use this algorithm for all examples and experiments.

Hyperparameters and Potential Issues

A rule that exactly matches x is always a valid anchor, albeit with very low coverage, and thus, is of little use. Our algorithm can always recover this anchor, and thus is guaranteed to terminate in a bounded number of iterations. In pathological cases, it is possible for KL-LUCB to require a very large number of samples from $\mathcal{D}$ in order to separate two candidate rules with a high confidence; however, this can be alleviated by increasing the tolerance ϵ, the width δ, or by setting a maximum number of samples. In practice, all explanations present in this paper were generated in a few seconds to few minutes. We set these parameters to reasonable values, $B = 10$, $\epsilon = 0.1$, $\delta = 0.05$, and leave an analysis of the sensitivity of our approach to these for future work.

So far, we focused on computing anchors for individual predictions. In order to gain a more complete understanding of how the model works, the user needs to examine multiple explanations. Instead of randomly selecting which anchors to show to the user, we would like to identify an *optimal* set of anchors that represent this global behavior, thereby reducing the user effort. By observing that such an objective is *submodular*, we propose an approach for *submodular-pick* (SP) in Ribeiro, Singh, and Guestrin (2016b) that we adapt to this setting. In particular, the approach selects K anchors that cover as many instances in the validation set as possible. We use an iterative, greedy method that provides guarantees on the quality of our solution, due to the submodular nature of the optimization (Krause and Golovin 2014).

Experiments

We evaluate anchor explanations for complex models on a number of tasks, primarily focusing on how they facilitate accurate predictions by users (simulated and human) on the behavior of the models on unseen instances. Code and the data for all the experiments is available at `https://github.com/marcotcr/anchor-experiments`.

Simulated Users

For simulated users, we use the tabular datasets previously mentioned (*adult*, *rcdv* and *lending*). Each dataset is split such that models are trained with the *training* set, explanations are produced for instances in the *validation* set, and evaluated on instances in the *test* set. For each dataset, we train three different models: logistic regression (**lr**), 400 gradient boosted trees (**gb**) and a multilayer perceptron with two layers of 50 units each (**nn**). We generate both linear LIME (Ribeiro, Singh, and Guestrin 2016b) and anchor explanations for them.

		Precision		Coverage	
		anchor	**lime-n**	**anchor**	**lime-t**
adult	logistic	95.6	81.0	10.7	21.6
	gbt	96.2	81.0	9.7	20.2
	nn	95.6	79.6	7.6	17.3
rcdv	logistic	95.8	76.6	6.8	17.3
	gbt	94.8	71.7	4.8	2.6
	nn	93.4	65.7	1.1	1.5
lending	logistic	99.7	80.2	28.6	12.2
	gbt	99.3	79.9	28.4	9.1
	nn	96.7	77.0	16.6	5.4

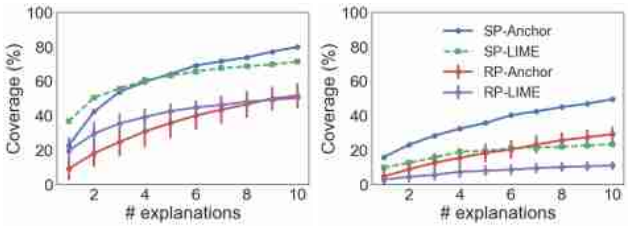
Table 4: Average precision and coverage with **simulated users** on 3 tabular datasets and 3 classifiers. *lime-n* indicates direct application of LIME to unseen instances, while *lime-t* indicates a threshold was tuned using an oracle to achieve the same precision as the anchor approach. The anchor approach is able to maintain very high precision, while a naive use of linear explanations leads to varying degrees of precision.

When simulating users, we compute coverage (what fraction of the instances they predict after seeing explanations) and precision (what fraction of the predictions were correct) on the complete test set. For each dataset, model, and explanation type, we compute these metrics for the explanation of each instance in the validation data. Simulating when an anchor *applies* is clear. It is not obvious, however, how real users would use LIME explanations. Ideally, they should only apply explanations to examples that are close, but it is not clear what the distance function and the threshold for "close" should be, or if users compute distances on demand. Therefore, in this section, we simulate different behaviors, and perform a study with real users in the following section.

In Table 4 (left), we show the average precision for anchor and LIME, assuming users always apply the LIME explanation without any regard to distance (we call such a user *lime-n*, for naive). It is clear that the anchor approach is able to deliver on the promise of high average precision, for all datasets and models. If users apply LIME naively, on the other hand, they get widely differing levels of precision. Since high precision is a prerequisite for interpretability, we simulate a user that only makes a prediction with LIME if the application of the linear explanation yields a probability above a certain threshold (setting a threshold on the distance produced strictly worse results), and call this user *lime-t*. We tune the threshold values for each dataset/model pair in order to obtain the same average precision as the anchor approach on the test set, so that coverage is comparable. A real user would not be able to perform this tuning, as (1) we are "cheating" by looking at the test set, and (2) the threshold values found range from 67% to 91%, and often with huge variation

Method	Precision				Coverage (perceived)				Time/pred (seconds)			
	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2
No expls	54.8	83.1	61.5	68.4	79.6	63.5	39.8	30.8	29.8 ±14	35.7±26	18.7±20	13.9±20
LIME(1)	68.3	98.1	57.5	76.3	89.2	55.4	71.5	54.2	28.5±10	24.6±6	8.6±3	11.1±8
Anchor(1)	100.0	97.8	93.0	98.9	43.1	24.6	31.9	27.3	13.0±4	14.4±5	5.4±2	3.7±1
LIME(2)	89.9	72.9	-	-	78.5	63.1	-	-	37.8±20	24.4±7	-	-
Anchor(2)	87.4	95.8	-	-	62.3	45.4	-	-	10.5±3	19.2±10	-	-

Table 5: **Results of the User Study.** Underline: significant w.r.t. anchors in the same dataset and same number of explanations. Results show that users consistently achieve high precision with anchors, as opposed to baselines, with less effort (time).



(a) *adult* dataset (b) *rcdv* dataset

Figure 4: Coverage on the test set as the simulated user sees more explanations, at the same precision level. While there is no clear winner for random explanations, anchors are better when explanations are picked using submodular-pick.

in the same dataset for different models. We show the average test coverage of the validation set explanations for anchor and lime-t in Table 4 (right). There is no clear winner in terms of coverage, thus demonstrating that even with the impossibly tuned thresholds, LIME is not able to outperform anchors.

Although the user is often interested in specific explanations, most users would prefer a set of explanations that explain most of the model with as little effort on their part as possible - explanations picked using the submodular procedure described before. In Figure 4, we show the coverage (for the same precision level) for *gb* in two of the datasets (we omit the other models/dataset due to space, but the results are similar) as the user sees more explanations, chosen either via submodular pick (SP-LIME and SP-Anchor) or at random (RP-LIME and RP-Anchor). The results indicate that while the average coverage of random explanations is low (e.g. 4.8% for *rcdv*), selecting the right set of explanations with submodular pick can give users an understanding of the model's *global* behavior (e.g. $\sim$50% after only 10 explanations). The anchor approach also yields better coverage for the same precision - even though it is unclear if real users can achieve such high precision with LIME.

User study

We ran a user study with 26 users – students who had or were taking a machine learning course – so we could rely on familiarity with concepts such as "model", "prediction", and "cross validation". For this study, we used the *adult* and *rcdv* datasets, followed by a multiple-choice VQA system (Ren, Kiros, and Zemel 2015) on two images. While the VQA model predicts one of 1000 labels, we restrict it to the 5 most common answers predicted on questions in $\mathcal{D}$, in order to reduce visual overload.

For each dataset and explanation type, we want to evaluate if users are able to predict the behavior of the model on unseen instances. Our set up for the *adult* and *rcdv* datasets consists of the users first browsing through 10 predictions without any explanations, then with one, and two LIME or anchor explanations. They are asked to predict the behavior of the classifier on 10 random test instances before and 10 instances after seeing each round of explanations. The user then goes through the same procedure on the other dataset, with the explanation type that was not the one used for the first one. We ask subjects to only make predictions if they are very confident, and to select "I don't know" otherwise. We measure coverage as the fraction of instances where users made a prediction other than "I don't know" (i.e. their perceived coverage), and only measure precision in these instances. This process is repeated for the two VQA images - half the users see LIME for the first and then anchor for the second, and vice versa for the other half, and predict the model's answers on 20 questions before and after explanations. The explanations are comparable in size: LIME explanations had at most 5 terms in all datasets, while anchors varied from 1 to 5, depending on each individual prediction.

The results in Table 5, where LIME(1) refers to results after one LIME explanation and so on, show that users with anchors achieve high-precision - around 95% for all of the combinations, except for Anchor(2) in *adult*. The coverage of users with anchors grows with a second explanation on both datasets. Precision with LIME, on the other hand, varies dramatically, and for VQA1, is worse than no explanations. The subjects made mostly correct predictions when using anchors (high precision), and knew to select "I don't know" when an instance was not covered by the explanation. In contrast, with LIME or no explanations, users thought they could make confident predictions more often, even though their precision was considerably lower. Further, it took dramatically less time to understand and use anchors as compared to linear explanations, across all datasets and tasks. On a poll, $21/26$ of users preferred anchors, and $24/26$ said they would be more precise with anchors; interestingly, the 2 users who said they would be more precise with LIME were actually more precise with anchors. Many commented that it was easier to

apply anchors than combining the weights of LIME explanations, especially with multiple explanations. They also felt more confident in their predictions with anchors.

The user study confirms our hypotheses: it is much easier for users to understand the coverage of anchor explanations as opposed to linear explanations, and to achieve high-precision understanding of the model's behavior (as measured by predicting it on new instances). Anchors are also easier to comprehend, and take less effort in applying, as reflected in their times and qualitative feedback.

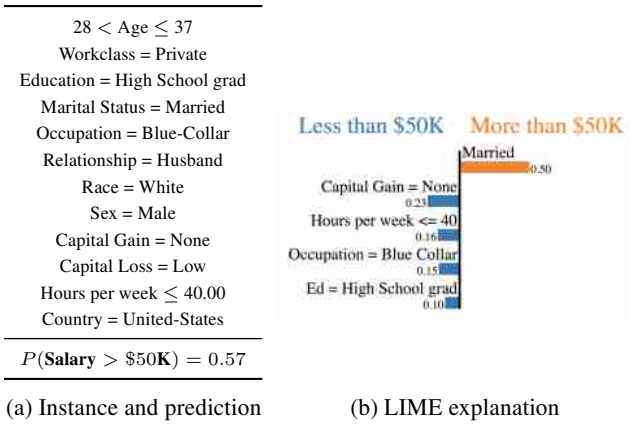
Limitations and Future Work

Having demonstrated the flexibility and usefulness of the anchor approach for a wide variety of domains, and having compared it to the state-of-the-art, we now turn to its limitations and opportunities for future work.

Overly specific anchors: Predictions that are near a boundary of the black box model's decision function, or predictions of very rare classes may require very specific "sufficient conditions", and thus their anchors may be complex and provide low coverage. We give an example in Figure 5 (*adult* dataset), where a particular prediction is very near the boundary of the decision function – almost any change to the instance results in a change in prediction. While the very specific anchor in Figure 5c communicates to the user that the conditions necessary for the prediction are very specific, it does not generalize well to other instances, due to its narrowness. It also does not give much insight into the model's behavior besides the fact that the instance is near a decision boundary. In cases like these, a linear LIME explanation (Figure 5b) may be preferred due to the potential insights it gives, with the caveat that users may still generalize incorrectly due to unclear coverage.

Potentially conflicting anchors: When using the anchor approach "in the wild", two or more anchors with different predictions may apply to the same test instance. While possible, this situation is unlikely for two reasons: (1) the high probability precision guarantee that anchors have by construction, and (2) the submodular objective in the pick procedure encourages a set of anchors with low overlap. If this were to happen in a real situation, we would want to alert the user, suggest further investigation, and maybe suggest increasing the precision threshold.

Complex output spaces: For certain problems where the output is structured and complex, there is a variety of explanations that may be useful. In this work, we restrict ourselves to explaining certain functions of the output, such as in Tables 1 and 2, leaving the task of explaining the full output space to future work. We emphasize that this is a problem that is not specific to the anchor approach, but still an open problem in the explanation literature. Even for a "simpler" output space such as the one present in the multi-label classification setting (Tsoumakas and Katakis 2006), it is not clear if the best option would be to explain each label individually or the set of predicted labels as a single label. The former could overwhelm the user if the number of labels is too large, while the latter may lead to non intuitive, or overly complex explanations.



(a) Instance and prediction

(b) LIME explanation

IF Country = United-States **AND** Capital Loss = Low
AND Race = White **AND** Relationship = Husband
AND Married **AND** 28 < Age ≤ 37
AND Sex = Male **AND** High School grad
AND Occupation = Blue-Collar
THEN PREDICT Salary > $50K

(c) An *anchor* explanation

Figure 5: Explaining a prediction near the decision boundary in the UCI adult dataset.

Realistic perturbation distributions: All perturbation-based explanation methods depend on a local perturbation distribution that is expressive enough to reveal the model's behavior, while acting on components that are sufficiently interpretable. Finding such distributions for some domains remains a challenge - for example, the perturbations for images provided explanations that led to some insight, but cannot be used for comparison of explanations across images. Designing such distributions is a line of research that would benefit multiple explanation methods.

Conclusions

We have argued that high precision and clear coverage are crucial for interpretable explanations of a model's local behavior. We introduced a novel family of rule-based, model-agnostic explanations called *anchors*, designed to exhibit both these properties. Anchors highlight the part of the input that is sufficient for the classifier to make the prediction, making them intuitive and easy to understand. We demonstrated the flexibility of the anchor approach by explaining predictions from a variety of classifiers on multiple domains. In a user study, we showed that anchors not only lead to higher human precision than linear explanations, but also require less effort to understand and apply.

Acknowledgements

We are grateful to the anonymous reviewers for their feedback. This work was supported in part by ONR award #N00014-13-1-0023, and in part by FICO and Adobe Research. The views expressed are those of the authors and do not reflect the policy or position of the funding agencies.

References

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K.-R. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11.

Cover, T. M., and Thomas, J. A. 1991. *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience.

Craven, M. W., and Shavlik, J. W. 1996. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems* 24–30.

De Raedt, L., and Kersting, K. 2008. Probabilistic inductive logic programming. Berlin, Heidelberg: Springer-Verlag. chapter Probabilistic Inductive Logic Programming, 1–27.

Kaufman, S.; Rosset, S.; and Perlich, C. 2011. Leakage in data mining: Formulation, detection, and avoidance. In *Knowledge Discovery and Data Mining (KDD)*.

Kaufmann, E., and Kalyanakrishnan, S. 2013. Information complexity in bandit subset selection. In *Proceedings of the Twenty-sixth nnual Conference on Learning Theory (COLT 2013)*, volume 30 of *JMLR Workshop and Conference Proceedings*, 228–251. JMLR.

Krause, A., and Golovin, D. 2014. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press.

Lakkaraju, H.; Bach, S. H.; and Leskovec, J. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1675–1684. New York, NY, USA: ACM.

Letham, B.; Rudin, C.; McCormick, T. H.; and Madigan, D. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*.

Lim, B. Y.; Dey, A. K.; and Avrahami, D. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, 2119–2128. New York, NY, USA: ACM.

Patel, K.; Fogarty, J.; Landay, J. A.; and Harrison, B. 2008. Investigating statistical machine learning as a tool for software development. In *Human Factors in Computing Systems (CHI)*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Ren, M.; Kiros, R.; and Zemel, R. S. 2015. Exploring models and data for image question answering. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, 2953–2961. Cambridge, MA, USA: MIT Press.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016a. Model-agnostic interpretability of machine learning. In *Human Interpretability in Machine Learning workshop*, ICML '16.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016b. "why should I trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*.

Sanchez, I.; Rocktaschel, T.; Riedel, S.; and Singh, S. 2015. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*.

Schmidt, P., and Witte, A. D. 1988. *Predicting Recidivism in North Carolina, 1978 and 1980*. Inter-university Consortium for Political and Social Research.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; and Crespo, J.-F. 2015. Hidden technical debt in machine learning systems. In *Neural Information Processing Systems (NIPS)*.

Strumbelj, E., and Kononenko, I. 2010. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research* 11.

Stumpf, S.; Rajaram, V.; Li, L.; Burnett, M.; Dietterich, T.; Sullivan, E.; Drummond, R.; and Herlocker, J. 2007. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI '07, 82–91. New York, NY, USA: ACM.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

Tsoumakas, G., and Katakis, I. 2006. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* 3(3).

Ustun, B., and Rudin, C. 2015. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*.

Vedaldi, A., and Soatto, S. 2008. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision*, 705–718. Springer.

Vinyals, O.; Kaiser, L.; Koo, T.; Petrov, S.; Sutskever, I.; and Hinton, G. E. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2773–2781.

Wang, F., and Rudin, C. 2015. Falling rule lists. In *Artificial Intelligence and Statistics (AISTATS)*.

Wang, T.; Rudin, C.; Doshi-Velez, F.; Liu, Y.; Klampfl, E.; and MacNeille, P. 2015. Or's of and's for interpretable classification, with application to context-aware recommender systems. *arXiv:1504.07614*.

Wieting, J.; Bansal, M.; Gimpel, K.; and Livescu, K. 2015. Towards universal paraphrastic sentence embeddings. *CoRR* abs/1511.08198.

Zhu, Y.; Groth, O.; Bernstein, M.; and Fei-Fei, L. 2016. Visual7W: Grounded Question Answering in Images. In *IEEE Conference on Computer Vision and Pattern Recognition*.