# R-LIME: Rectangular Constraints and Optimization for Local Interpretable Model-agnostic Explanation Methods

Genji Ohara[0009−0000−5854−2820], Keigo Kimura[0000−0002−3614−6568], and Mineichi Kudo[0000−0003−1013−3870]

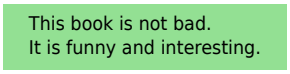
Division of Computer Science and Information Technology
Graduate School of Information Sci. and Tech., Hokkaido University
Sapporo 060-0814, JAPAN,
{genji-ohara, kimura5, mine}@ist.hokudai.ac.jp

**Abstract.** In recent years, complex machine learning models such as deep neural networks have been used in various industrial fields due to their high accuracy. However, its complexity has been a major obstacle to implementation in decision-making situations where transparency of the decision process is required. In order to address this problem, various post-hoc explanation methods have been proposed, but they have not been able to achieve interpretability of both the explanation and its scope. We propose a new method, R-LIME, which interprets complex classifiers in an interpretable scope. R-LIME locally approximizes a complex decision boundary linearly in a rectangular region and maximizes the region as long as the accuracy of the linear classifier is higher than a specified threshold. The resulting rectangular region is interpretable for users because it is expressed as a conjunction of feature predicates. We demonstrate the effectiveness of the proposed method through qualitative and quantitative experiments using a real-world dataset. Finally, we discuss limitations and future work of the proposed method.

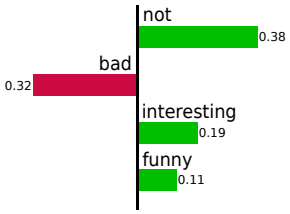
**Keywords:** Interpretable machine learning · Local surrogate model

## 1 Introduction

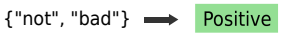
Machine learning models, such as deep learning and random forests, have been widely employed in various industrial applications due to their significant improvement in accuracy in recent years. However, the increasing complexity and black-box nature of these models pose challenges, particularly in critical decision-making scenarios like healthcare and finance, where the opacity of decision process becomes a major obstacle to implementation. Consequently, there has been extensive research in the field of post-hoc explanations for machine learning models [4, 8–10]. Existing post-hoc explanation methods are categorized into *model-dependent* and *model-agnostic* methods based on their dependence on the model's structure. Furthermore, model-agnostic methods are classified into *global* and *local* methods based on their locality in input space.



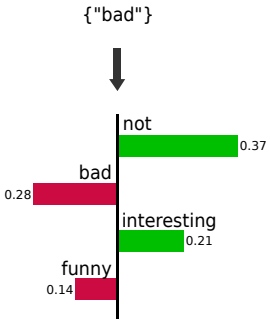

(a) Focal point.




(b) Explanation by LIME. It provides contributions of each feature to the output, but does not explicitly indicate its scope.




(c) Explanation by Anchor. It provides the effective scope of the explanation, but does not show the influence of each feature.




(d) Explanation by R-LIME. It provides both contributions of each feature and its effective scope.

Fig. 1: Example of explanations by LIME [9], Anchor [10] and R-LIME (our proposed method). for a sentiment prediction model.

In this paper, we focus on local model-agnostic methods. An example of explanations by LIME [9] and Anchor [10] for a sentiment prediction model is illustrated in Fig. 1. LIME approximates the complex decision boundary linearly around the given focal point, then provides the weights of the linear model as the contribution of each feature to the output. The explanation by LIME (Fig. 1(b)) provides suggests that the word "not" mainly contributes to the positive prediction, but does not explicitly indicate the effective scope. Without the scope, users might mistakenly apply the knowledge derived from the explanation to other instances far from the focal point, potentially leading to misunderstanding of the black-box model's behavior [10]. For this example, user may apply the derived insights to the sentence "This book is not good." and mistakenly conclude that the word "not" mainly contributes to the positive prediction for this sentence as well, which is obviously incorrect. Anchor represents a rectangular region containing the focal point, that maximizes the probability of the black-box classifier outputting the same label as the focal point. While Anchor provides an effective scope of the explanation, the information it includes is less than LIME, limiting its utility [10]. The explanation by Anchor (Fig. 1(c)) suggests that changing words other than "not" and "bad" has little impact on the classifier's output. While it clearly cannot be applied to the sentence "This book is not good" because of not including the word "bad", the explanation does not provide details about the influence of each word, resulting in less user insight into the model's behavior compared to LIME.

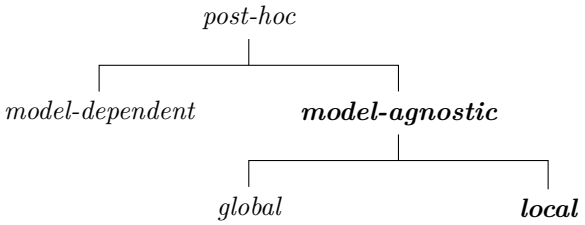



Fig. 2: Categorization of post-hoc explanation methods. We focus on *model-agnostic* and *local* methods, which explain model's local behavior using only its output.

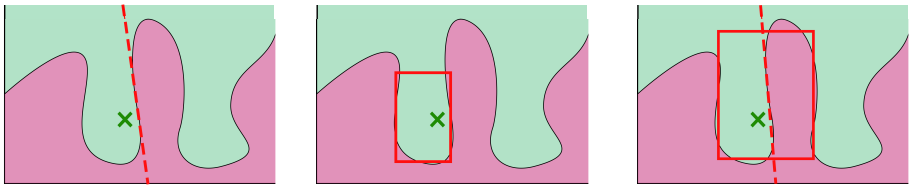
To address this limitation of LIME, we propose a new method called R-LIME (Ruled LIME), which interprets complex classifiers in an interpretable region. R-LIME locally approximates a complex decision boundary linearly in a rectangular region and maximizes the region as long as the accuracy of the linear classifier is higher than a given threshold. The rectangular region is interpretable for users because it is expressed as a conjunction of feature predicates (Fig. 1(d)). An example of the explanation by R-LIME for a sentiment prediction model is illustrated in Fig. 1(d). It is clear that users can apply the insights derived from the explanation only to the sentences containing the word "not".

## 2 Related Work

In this chapter, we overview existing research on post-hoc explanation methods, which explain the behavior of black-box models already trained. As shown in Fig. 2, post-hoc methods are classified into several categories.

First, they are categorized into *model-dependent* and *model-agnostic* methods based on their dependence on the model's structure. Model-dependent methods, such as Deep Taylor Decomposition (DTD) [7] and Layer-wise Relevance Propagation (LRP) [2], most of which focus on neural networks and explain the model's behavior using its parameters [11]. While these methods provide detailed explanations (e.g., layer-wise explanations for neural networks), it is often challenging to apply the same method to models with different structures. On the other hand, model-agnostic methods use only the output of the model. Although they are applicable to any model, they cannot explain the reasoning process inside the model.

Then, model-agnostic methods are classified into *global* and *local* methods based on their locality in input space. Global methods, such as Partial Dependence Plots (PDP) [3] and Accumulated Local Effects (ALE) [1], aim to explain the model's behavior across the entire input space. However, providing global explanations becomes challenging as the model's complexity increases. In contrast, local methods, such as Local Interpretable Model-agnostic Explanations (LIME) [9] and Anchor [10], explain model's behavior in the vicinity of a specific input. While local methods offer explanations more simple and accurate than global methods, the scope of the explanation is limited locally.



(a) LIME: Locally approximates the decision boundary around the focal point.

(b) Anchor: Maximizes coverage of a rectangular region containing the focal point under accuracy constraints.

(c) R-LIME: Maximizes coverage of a rectangular region under lower constraints on the accuracy of the linear classifier.

Fig. 3: Visual comparison of LIME, Anchor and R-LIME (our proposed method). The solid line represents the rectangular region containing the focal point, and the dashed line represents the learned approximation model.

## 3 Proposed Method

### 3.1 Previous Work

We specifically focuses on *local* and *model-agnostic* methods. In this section, we briefly review existing research on local model-agnostic explanations, particularly focusing on studies closely related to our proposed method.

**LIME (Local Interpretable Model-agnostic Explanations) [9]** LIME locally approximates a black-box classifier $f : \mathbb{R}^m \to \{0, 1\}$ around a focal point $x \in \mathbb{R}^m$ by a linear classifier $g : \mathbb{R}^m \to \{0, 1\}$ (Fig. 3(a)). The approximation is performed by the following two steps:

1. Generating a set of perturbed samples $\mathcal{Z}_p$ around $x$ and the set of pseudo-labels $f(\mathcal{Z}_p) = \{f(z) \mid z \in \mathcal{Z}_p\}$. (i) $x$ is converted into a binary vector $x' \in \{0, 1\}^{m'}$, (ii) perturbed samples are generated by drawing non-zero elements from $x'$ uniformly at random, and (iii) the perturbed samples are converted back to the original space.
2. Learning a linear classifier $g$ using $\mathcal{Z}_p$ and $f(\mathcal{Z}_p)$ by minimizing the following loss function:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}_p} \pi_x(z)(f(z) - g(z))^2, \tag{1}$$

   where $\pi_x(z)$ is a weight function designed to be larger for samples closer to $x$, typically implemented using an exponential kernel.

LIME provides valuable insights into the local behavior of the model by showing the contribution of each feature to the output $f(x)$. However, due to not explicitly indicating the perturbation region, users cannot assess the effective scope of the explanation [10].

**Anchor [10]** Anchor represents a rectangular region containing the focal point $x$, expressed as a conjunction of feature predicates (a rule), that maximizes the probability of the black-box classifier $f$ outputting $f(x)$ within the region. It aims to highlight important features contributing significantly to the output (Fig. 3(b)). For a discrete $m$-dimensional input space $\mathbb{D}^m$ with a trained black-box classifier $f : \mathbb{D}^m \to \{0, 1\}$, an instance $x \in \mathbb{D}^m$, and a distribution $\mathcal{D}$ over the input space, a rule $A(z) = a_{i_1}(z) \wedge a_{i_2}(z) \wedge \cdots \wedge a_{i_t}(z)$ is defined. The predicates $a_i(z)$ evaluate to true (= 1) when $z_i = x_i$ and false (= 0) otherwise. The accuracy $\mathrm{acc}(A)$ and coverage $\mathrm{cov}(A)$ of the rule $A$ are defined as follows:

$$\mathrm{acc}(A) = \mathbb{E}_{z \sim \mathcal{D}(z|A)}[\mathbb{1}_{f(z)=f(x)}], \tag{2}$$

$$\mathrm{cov}(A) = \mathbb{E}_{z \sim \mathcal{D}(z)}[A(z)]. \tag{3}$$

where $\mathcal{D}(z|A)$ is the conditional distribution in the region where the rule $A$ returns true. $\mathrm{acc}(A)$ represents the probability that the output of $f$ matches between the perturbation $z \sim \mathcal{D}(z|A)$ and the focal point $x$, and $\mathrm{cov}(A)$ expresses the probability that the perturbation $z$ fits into $A$. Anchor maximizes coverage under the constraint that the accuracy of the rule $A$ exceeds a given threshold $\tau$. However, eq. (2) is not directly computable. Introducing a confidence level $1 - \delta$ $(0 \leq \delta \leq 1)$, the accuracy constraint is relaxed as follows:

$$P(\mathrm{acc}(A) \geq \tau) \geq 1 - \delta. \tag{4}$$

Thus, the following optimization problem is solved:

$$A^* = \underset{A \text{ s.t. } P(\mathrm{acc}(A) \geq \tau) \geq 1-\delta \wedge A(x)=1}{\arg\max} \mathrm{cov}(A). \tag{5}$$

### 3.2 Overview

We propose R-LIME, a method that aims to address the limitations of LIME [9], and Anchor [10]. Our method locally approximates the given black-box classifier $f$ around the focal point $x$ by a linear classifier $g$ similar to LIME and BELLA, but samples perturbation from a rectangular region similar to Anchor so that the generality of approximation is explicitly provided (Fig. 3(c)). The accuracy of $g$ is defined as the "accuracy" of the rule, and the generality of explanations is defined as the "coverage" of the rule.

Anchor maximizes the coverage of region $A$ as long as the probability of the output of the black-box classifier $f$ matching $f(x)$ within $A$ exceeds a given threshold $\tau$. The proposed method, on the other hand, learns an linear classifier $g$ within the rectangular region $A$ and maximizes the coverage of $A$ under lower constraints on the accuracy of $g$. We modify the Anchor's definition of accuracy in eq. (2) as follows:

$$\mathrm{acc}(A) = \max_{g \in G} \mathbb{E}_{z \sim \mathcal{D}(z|A)}[\mathbb{1}_{f(z)=g(z)}]. \tag{6}$$

where $G$ is the set of possible linear classifiers. By solving the optimization problem in eq. (5) under the modified accuracy definition in eq. (6), we can select the rule that enables explanation with high accuracy and generality.

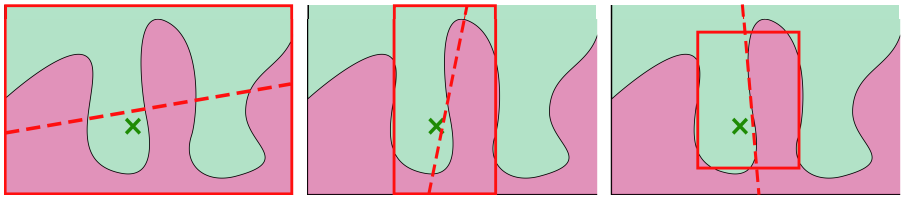


Fig. 4: Overview of the R-LIME algorithm. The progression of the algorithm is illustrated from left to right. The solid line represents the rectangular region $A$, and the dashed line represents the linear approximation model $g$ learned within $A$. The initial value of $A$ is an empty rule (entire input space), and predicates are added to $A$, reducing coverage. The process continues until $\mathrm{acc}(A) \geq \tau$, at which point the rule with the maximum coverage is output.

### 3.3 Algorithm

The algorithm of R-LIME is mainly based on that used in Anchor[10]. For non-convex optimization problems like eq. (5), greedy search are often used. However, greedy methods tend to converge to local optima, and to address this, R-LIME utilizes beam search, which selects multiple candidates at each iteration. The pseudo code is shown in Algorithm 1.

**Generation of New Candidate Rules** To generate new candidate rules, one additional predicate is added to each of the $B$ candidate rules selected in the previous iteration. The pseudo code is shown in Algorithm 2. $T(x)$ is the set of attribute-value pairs (predicates) that are true for $x$, and $T(x) \setminus A$ is the set of predicates in $T(x)$ not included in rule $A$.

**Selection of Candidate Rules with Maximum Accuracy** Given the set of candidate rules $\bar{\mathcal{A}}$, the algorithm selects the top $B$ candidates with the highest accuracy. This problem can be formulated as best arm identification in the multi-armed bandit framework. Each candidate rule $A_i \in \bar{\mathcal{A}}$ is considered as an arm, and its accuracy $\mathrm{acc}(A_i)$ is considered as the distribution of rewards. By sampling $z \sim \mathcal{D}(\cdot|A_i)$ and obtaining the reward $\mathbb{1}_{f(z)=g_i(z)}$ for each trial, the algorithm updates $g_i$ using the sampled perturbation vector $z$ and the pseudo-label $f(z)$ after each trial. To efficiently select the rule (arm) with the highest accuracy, we employ the KL-LUCB algorithm [5]. The pseudo code is shown in Algorithm 3. For tolerance $\epsilon \in [0, 1]$, the KL-LUCB algorithm guarantees below:

$$P(\min_{A \in \bar{\mathcal{A}}} \mathrm{acc}(A) \geq \min_{A' \in \mathcal{A}} \mathrm{acc}(A') - \epsilon) \geq 1 - \delta. \tag{7}$$

However, the KL-LUCB algorithm assumes that the reward distribution for each arm remains unchanged, while our method updates the classifier $g_i$ with each sampling, which may not satisfy this assumption. This issue is discussed further in section 5.2.

**Algorithm 1** R-LIME

**Input:** Black-box model $f$, Target instance $x$, Distribution $\mathcal{D}$, Threshold $\tau$, Beam width $B$, Tolerance $\epsilon$, Confidence level $1-\delta$
**Output:** Rule $A^*$ satisfying Eq. (5)

$A^* \leftarrow$ **null**, $\mathcal{A}_0 \leftarrow \emptyset$, $t \leftarrow 0$ ▷ Initialize the set of candidate rules $\mathcal{A}_0$ to $\emptyset$
**while** $A^* =$ **null do**
  $t \leftarrow t+1$
  $\bar{\mathcal{A}}_t \leftarrow$ GenerateCands$(\mathcal{A}_{t-1})$ ▷ Generate new candidate rules from $\mathcal{A}_{t-1}$
  $\mathcal{A}_t \leftarrow$ B-BestCands$(\bar{\mathcal{A}}_t, \mathcal{D}, B, \epsilon, \delta)$ ▷ Select $B$ best candidate rules from $\bar{\mathcal{A}}_t$
  $A^* \leftarrow$ LargestCand$(\mathcal{A}_t, \tau, \delta)$ ▷ Select the largest rule satisfying Eq. (5)
**end while**

**Algorithm 2** Generating candidate rules

**function** GenerateCands$(\mathcal{A}, x)$
  **if** $\mathcal{A} = \emptyset$ **then return** $\{\emptyset\}$
  $\bar{\mathcal{A}} \leftarrow \emptyset$
  **for all** $A \in \mathcal{A}$ **do**
    **for all** $a_i \in (T(x) \setminus A)$ **do**
      $\bar{\mathcal{A}} \leftarrow \bar{\mathcal{A}} \cup (A \wedge a_i)$
    **end for**
  **end for**
  **return** $\bar{\mathcal{A}}$
**end function**

**Algorithm 3** Selecting $B$ best rules with highest accuracy (KL-LUCB [5])

**function** B-BestCands$(\bar{\mathcal{A}}, \mathcal{D}, B, \epsilon, \delta)$
  **initialize** $\mathrm{acc}, \mathrm{acc}_u, \mathrm{acc}_l$ for $\forall A \in \bar{\mathcal{A}}$
  $\mathcal{A} \leftarrow$ B-ProvisionallyBestCands$(\bar{\mathcal{A}})$ ▷ $B$ rules with highest accuracy
  $A \leftarrow \arg\min_{A \in \mathcal{A}} \mathrm{acc}_l(A, \delta)$ ▷ The rule with the smallest lower bound
  $A' \leftarrow \arg\max_{A' \notin (\bar{\mathcal{A}} \setminus \mathcal{A})} \mathrm{acc}_u(A', \delta)$ ▷ The rule with the largest upper bound
  **while** $\mathrm{acc}_u(A', \delta) - \mathrm{acc}_l(A, \delta) > \epsilon$ **do**
    **sample** $z \sim \mathcal{D}(z|A), z' \sim \mathcal{D}(z'|A')$
    **update** $\mathrm{acc}, \mathrm{acc}_u, \mathrm{acc}_l$ for $A$ and $A'$
    $\mathcal{A} \leftarrow$ B-ProvisionallyBestCands$(\bar{\mathcal{A}})$
    $A \leftarrow \arg\min_{A \in \mathcal{A}} \mathrm{acc}_l(A, \delta)$
    $A' \leftarrow \arg\max_{A' \notin (\bar{\mathcal{A}} \setminus \mathcal{A})} \mathrm{acc}_u(A', \delta)$
  **end while**
  **return** $\mathcal{A}$
**end function**

**Algorithm 4** Selecting the largest rule satisfying the constraint

**function** LargestCand$(\mathcal{A}, \tau, \delta)$
  $A^* \leftarrow$ **null**
  **for all** $A \in \mathcal{A}$ s.t. $\mathrm{acc}_l(A, \delta) > \tau$ **do**
    **if** $\mathrm{cov}(A) > \mathrm{cov}(A^*)$ **then** $A^* \leftarrow A$
  **end for**
  **return** $A^*$
**end function**

**Selection of the Rule with Maximum Coverage Meeting the Constraint** To satisfy the constraint imposed by eq. (4), rule $A$ needs to fulfill the lower bound constraint:

$$\mathrm{acc}_l(A, \delta) > \tau \tag{8}$$

where $\mathrm{acc}_l(A, \delta)$ is the lower limit of the $100(1-\delta)\%$ confidence interval for $\mathrm{acc}(A)$. If the received set of candidate rules, $\mathcal{A}$, includes a rule satisfying eq. (8), the one with the maximum coverage among them is selected, and the iteration is terminated. If $\mathcal{A}$ does not contain any rule satisfying eq. (8), it returns **null**, and proceeds to the next iteration. The pseudo code is presented in Algorithm 4.

### 3.4 Computational Complexity

Post-hoc explanation methods including LIME, Anchor, and R-LIME need to query the black-box model multiple times to get labels for perturbed samples, which is computationally expensive. The number of queries required for LIME is $|\mathcal{Z}_p|$, where $|\mathcal{Z}_p|$ is the number of perturbed samples selected by the user. On the other hand, the number of queries required for Anchor and R-LIME is $\mathcal{O}(B \cdot m^2 + m^2 \cdot \mathcal{O}_{\mathrm{MAB}[B \cdot m, B]})$, where $B$ is the beam width, $m$ is the number of features, and $\mathcal{O}_{\mathrm{MAB}[B \cdot m, B]}$ is the computational complexity of best arm identification finding the $B$ best arms among $B \cdot m$ arms. $\mathcal{O}_{\mathrm{MAB}[B \cdot m, B]}$ depends on hyperparameters such as beam width $B$, confidence level $1 - \delta$ and tolerance $\epsilon$.

## 4 Experiments

To verify the effectiveness of the proposed method, We compare LIME and R-LIME using a real-world dataset.

### 4.1 Qualitative Evaluation

**Experimental Setup** The experiments utilized the recidivism dataset [12]. This dataset contains information on 9,549 prisoners released from North Carolina prisons between July 1, 1979, and June 30, 1980. The dataset includes 19 items such as race, gender, presence of alcohol dependence, number of prior offenses and recidivism. For this experiment, we treated the binary classification problem of predicting the presence or absence of recidivism (Recidivism) as the target label. We discretized continuous features and removed missing values, resulting in 15 features. This problem setting can be considered as a case where a machine learning model is introduced to decide parole for prisoners. Since such decisions can have a significant impact on a person's life, it is crucial for users to interpret the outputs of black-box models appropriately.

The dataset, after removing missing values, was split into training data (7,639 instances) and test data (955 instances). A random forest model with 50 trees was trained using the training data. Subsequently, LIME and R-LIME explanations were generated for two instances extracted from the test data (Fig. 5). For

Table 1: Attributes of the recidivism dataset used in the experiments. Continuous features are all discretized, and only binary and ordinal features are considered.

| Attribute | Overview | # of Possible Values |
|---|---|---|
| Race | Race (Black or White) | 2 |
| Alcohol | Presence of serious alcohol issues | 2 |
| Junky | Drug usage | 2 |
| Supervised Release | Supervised release | 2 |
| Married | Marital status | 2 |
| Felony | Felony or not | 2 |
| WorkRelease | Participation in work release program | 2 |
| Crime against Property | Crime against property or not | 2 |
| Crime against Person | Crime against a person or not | 2 |
| Gender | Gender (Female or Male) | 2 |
| Priors | Number of prior offenses | 4 |
| YearsSchool | Years of formal education completed | 4 |
| PrisonViolations | Number of prison rule violations | 3 |
| Age | Age | 4 |
| MonthsServed | Months served in prison | 4 |
| Recidivism | Recidivism or not | 2 |

R-LIME, logistic regression was used as the linear approximation model, and the distribution $\mathcal{D}$ was a multivariate normal distribution estimated from the training data. The beam width was set to $B = 10$, the confidence coefficient to $1-\delta = 0.95$, and the tolerance of the KL-LUCB algorithm to $\epsilon = 0.05$. Accuracy thresholds $\tau$ were set to $\tau = 0.70, 0.80, 0.90$.

**Experimental Results** The results of the experiment are shown in Figs. 6 and 7. The values assigned to each attribute name represent the contribution (weights of the learned linear classifier) to the output of the black-box classifier, normalized such that the absolute sum is 1. The figures also display the 5 attributes with the highest absolute contribution.

Explanations generated by LIME (Figs. 6(a) and 7(a)) provide insights that attributes such as having a prior offenses (Priors), being served for a long time in prison (MonthsServed), and committing a crime against property (Crime against Property) primarily contribute to the positive prediction (prediction that the prisoner will be re-arrested). On the other hand, attributes like older age (Age), being married (Married), and being of white race (Race) contribute to the negative prediction (prediction that the prisoner will not be re-arrested). While these LIME explanations provide valuable insights into the behavior of the black-box model, they do not explicitly indicate the application scope of the explanations, leaving users unable to determine to which inmates the explanations are applicable.

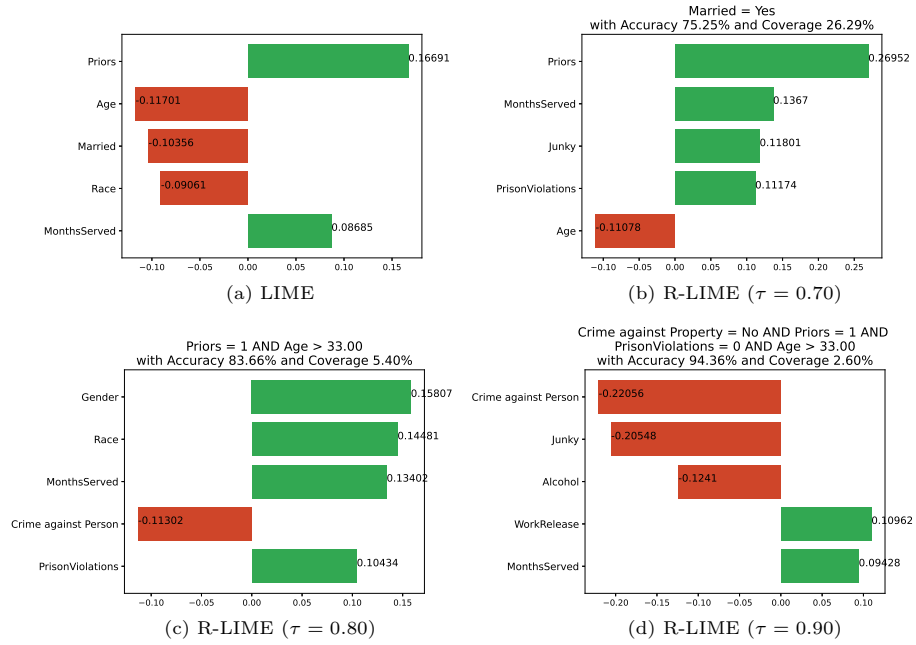
| | |
|---|---|
| Race | Black (0) |
| Alcohol | No (0) |
| Junky | No (0) |
| Supervised Release | Yes (1) |
| Married | Yes (1) |
| Felony | No (0) |
| WorkRelease | Yes (1) |
| Crime against Property | No (0) |
| Crime against Person | No (0) |
| Gender | Male (1) |
| Priors | 1 |
| YearsSchool | 8.00 < YearsSchool <= 10.00 (1) |
| PrisonViolations | 0 |
| Age | Age > 33.00 (3) |
| MonthsServed | 4.00 < MonthsServed <= 9.00 (1) |
| Recidivism | No more crimes (0) |

(a) Instance A

| | |
|---|---|
| Race | Black (0) |
| Alcohol | Yes (1) |
| Junky | No (0) |
| Supervised Release | Yes (1) |
| Married | No (0) |
| Felony | No (0) |
| WorkRelease | Yes (1) |
| Crime against Property | Yes (1) |
| Crime against Person | No (0) |
| Gender | Male (1) |
| Priors | 1 |
| YearsSchool | YearsSchool > 11.00 (3) |
| PrisonViolations | 0 |
| Age | 21.00 < Age <= 26.00 (1) |
| MonthsServed | 4.00 < MonthsServed <= 9.00 (1) |
| Recidivism | Re-arrested (1) |

(b) Instance B

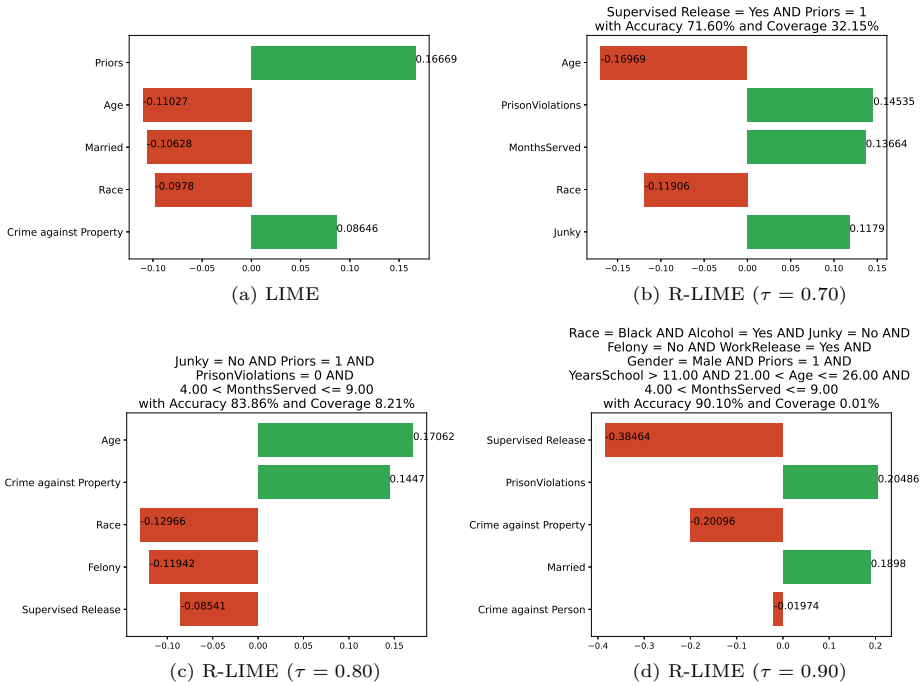
Fig. 5: Two instances sampled from recidivism dataset.




(a) LIME (b) R-LIME ($\tau = 0.70$)

(c) R-LIME ($\tau = 0.80$) (d) R-LIME ($\tau = 0.90$)

Fig. 6: Explanation for Instance A by LIME and R-LIME.




(a) LIME (b) R-LIME ($\tau = 0.70$)

(c) R-LIME ($\tau = 0.80$) (d) R-LIME ($\tau = 0.90$)

Fig. 7: Explanation for Instance B by LIME and R-LIME.

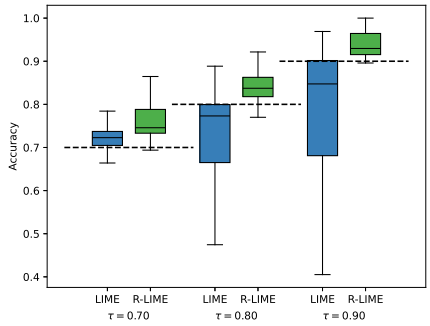


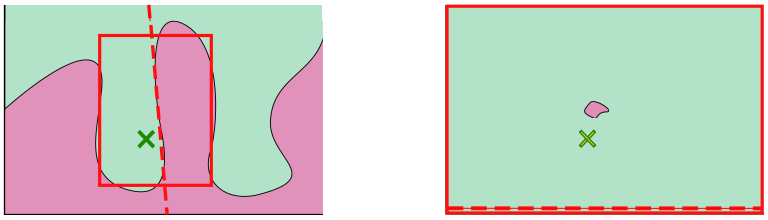

Fig. 8: Comparison of local accuracy between LIME and R-LIME. R-LIME achieved higher and less variable accuracy compared to LIME for all values of $\tau$.

In contrast, R-LIME expresses the application scope of explanations as a conjunction of predicates. For example, the explanation for instance A under $\tau = 0.70$ (Fig. 6(b)) indicates that it is applicable only to married prisoner (Married=Yes). Furthermore, the explanations include the accuracy and coverage, allowing users to evaluate its reliability. For example, the coverage of the explanation for instance B under $\tau = 0.90$ (Fig. 7(d)) is 0.01%, indicating that the decision boundaries around instance B are complex, making it challenging to obtain a high-accuracy linear approximation. This allows users to discern that the application scope of this explanation is very narrow, limiting its utility.

### 4.2 Quantitative Evaluation

**Experimental Setup** To demonstrate that R-LIME learns a highly accurate linear approximation model in the optimized approximation region, we conducted a comparison of the local accuracy of explanations between LIME and R-LIME. Using the same settings as in section 4.1, we randomly sampled 100 instances from the test data of the recidivism dataset and generated explanations (with $\tau = 0.70, 0.80, 0.90$) using LIME and R-LIME. We then sampled 10,000 instances within the rectangular region obtained by R-LIME and calculated the local accuracy of both methods.

**Experimental Results** The results are presented in Fig. 8, showing the distribution of the accuracy of the linear approximation models learned by LIME and R-LIME. R-LIME achieved higher accuracy compared to LIME for all values of $\tau$. This suggests that the linear classifiers learned by LIME and R-LIME differ significantly, and R-LIME learns a high-accuracy linear classifier adapted to the rectangular region. Additionally, as $\tau$ increases, the variability in the accuracy of LIME widens. This indicates that the linear classifiers learned by LIME may not function effectively as approximation models depending on how the region is selected.



(a) R-LIME for balanced label distribution. (b) R-LIME for imbalanced label distribution.

Fig. 9: Behavior of R-LIME for balanced and imbalanced label distribution. In case of imbalanced label distribution, the approximation region covers the entire input space and the linear approximation model always outputs the majority label.

Table 2: Deviation between the estimated accuracy and the true accuracy. Deviation was relatively small considering confidence level $1 - \delta = 0.95$.

| | Estimated acc. | True acc. | Deviation |
|---|---|---|---|
| Average | .811 | .829 | .012 |
| Standard Deviation | .018 | .023 | .017 |

## 5 Challenges and Future Perspectives

### 5.1 Behavior Regarding Imbalanced Label Distribution

R-LIME may generate less useful explanations when there is bias in the distribution of black-box model outputs. When the distribution of black-box model outputs is significantly biased for a given accuracy threshold $\tau$ (when the ratio of the minority label is less than $1 - \tau$), the approximation region generated by R-LIME covers the entire input space, and the learned linear classifier always outputs the majority label (Fig. 9(b)).

A first possible solution to this problem is modifying the loss function. Using weighted logistic loss or Focal Loss [6] as the loss function might lead to the generation of more useful explanations in the case of imbalanced label distribution. Another solution involves adding constraints to limit the label distribution bias within the approximation region. In addition to eq. (4), adding a constraint like

$$\left(\mathbb{E}_{z\sim\mathcal{D}(z|A)}[\mathbb{1}_{f(z)=1}] - \frac{1}{2}\right)^2 < \mu \tag{9}$$

could suppress the excessive expansion of the approximation region.

### 5.2 Changes in Reward Distribution in Best Arm Identification

For R-LIME, the problem of selecting the rule with the highest accuracy is formulated as the optimal arm identification problem in multi-armed bandit theory,

solved using the KL-LUCB algorithm [5]. However, this algorithm assumes that the reward distribution remains constant, while in R-LIME, the reward distribution (accuracy of the linear approximation) changes with every update of the approximation model after sampling. Therefore, rewards obtained at an early stage might influence the estimated value and deviate from the true value.

We conducted an experiment to evaluate the deviation between the estimated accuracy and the true accuracy. We generated explanations for 3,200 data instances sampled from the dataset, and compared the estimated accuracy with the true accuracy. The true accuracy was calculated based on 1,000 instances sampled within the approximation region. The results in Table 2 show a mean deviation of 0.012 with a standard deviation of 0.017. While the deviation was relatively small considering confidence level $1 - \delta = 0.95$, we should modify the selection algorithm to consider the varying accuracy.

## 6 Conclusion

We identified challenges in existing methods for local model-agnostic post-hoc explanations of black-box classifiers and proposed R-LIME to address them. We represented the rectangular region for local linear approximation as a conjunction of feature predicates and proposed an algorithm to maximize coverage under the constraint of minimum approximation accuracy. Comparing the outputs of LIME and R-LIME on the real-world dataset, we demonstrated that R-LIME provides a clear application scope of the explanation and can be evaluated by users for its reliability and generality. However, we discussed the instability of behavior concerning imbalanced label distributions and raised questions about the theoretical validity of using the KL-LUCB algorithm.

## References

1. Apley, D.W., Zhu, J.: Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. Journal of the Royal Statistical Society Series B: Statistical Methodology **82**(4), 1059–1086 (2020). https://doi.org/10.1111/rssb.12377, https://doi.org/10.1111/rssb.12377
2. Bach, Sebastian AND Binder, Alexander AND Montavon, Grégoire AND Klauschen, Frederick AND Müller, Klaus-Robert AND Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE **10**(7), 1–46 (2015). https://doi.org/10.1371/journal.pone.0130140, https://doi.org/10.1371/journal.pone.0130140
3. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics **29**(5), 1189–1232 (2001), http://www.jstor.org/stable/2699986
4. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems (2018)
5. Kaufmann, E., Kalyanakrishnan, S.: Information complexity in bandit subset selection. In: Shalev-Shwartz, S., Steinwart, I. (eds.) Proceedings of the 26th Annual Conference on Learning Theory. Proceedings of Machine Learning Research,

vol. 30, pp. 228–251. PMLR, Princeton, NJ, USA (12–14 Jun 2013), https://proceedings.mlr.press/v30/Kaufmann13.html
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 318–327 (2020). https://doi.org/10.1109/TPAMI.2018.2858826
7. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition **65**, 211–222 (2017). https://doi.org/https://doi.org/10.1016/j.patcog.2016.11.008, https://www.sciencedirect.com/science/article/pii/S0031320316303582
8. Radulovic, N., Bifet, A., Suchanek, F.: BELLA: Black box model explanations by local linear approximations (2023)
9. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778, https://doi.org/10.1145/2939672.2939778
10. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1), 1527–1535 (Apr 2018). https://doi.org/10.1609/aaai.v32i1.11491, https://ojs.aaai.org/index.php/AAAI/article/view/11491
11. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE **109**(3), 247–278 (2021). https://doi.org/10.1109/JPROC.2021.3060483
12. Schmidt, P., Witte, A.D.: Predicting Recidivism in North Carolina, 1978 and 1980. Inter-university Consortium for Political and Social Research (1988)