Probabilistic Explanations for Linear Models*

Bernardo Subercaseaux[1], **Marcelo Arenas**[2, 3, 4], **Kuldeep S. Meel**[5, 6]

[1]Carnegie Mellon University
[2]Pontificia Universidad Católica de Chile
[3]IMFD Chile
[4]RelationalAI
[5]Georgia Institute of Technology
[6]University of Toronto

Abstract

Formal XAI is an emerging field that focuses on providing explanations with mathematical guarantees for the decisions made by machine learning models. A significant amount of work in this area is centered on the computation of "sufficient reasons". Given a model $\mathcal{M}$ and an input instance $\boldsymbol{x}$, a sufficient reason for the decision $\mathcal{M}(\boldsymbol{x})$ is a subset S of the features of $\boldsymbol{x}$ such that for any instance $\boldsymbol{z}$ that has the same values as $\boldsymbol{x}$ for every feature in S, it holds that $\mathcal{M}(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{z})$. Intuitively, this means that the features in S are sufficient to fully justify the classification of $\boldsymbol{x}$ by $\mathcal{M}$. For sufficient reasons to be useful in practice, they should be as small as possible, and a natural way to reduce the size of sufficient reasons is to consider a probabilistic relaxation; the probability of $\mathcal{M}(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{z})$ must be at least some value $\delta \in (0, 1]$, for a random instance $\boldsymbol{z}$ that coincides with $\boldsymbol{x}$ on the features in S. Computing small δ-sufficient reasons (δ-SRs) is known to be a theoretically hard problem; even over decision trees — traditionally deemed simple and interpretable models — strong inapproximability results make the efficient computation of small δ-SRs unlikely. We propose the notion of (δ, ϵ)-SR, a simple relaxation of δ-SRs, and show that this kind of explanations can be computed efficiently over linear models.

Extended version — https://arxiv.org/abs/2501.00154

1 Introduction

Explaining the decisions of Machine Learning classifiers is a fundamental problem in XAI (Explainable AI), and doing so with formal mathematical guarantees on the quality, size, and semantics of the explanations is in turn the core of *Formal XAI* (Marques-Silva and Ignatiev 2022). Within formal XAI, one of the most studied kinds of explanations is that of *sufficient reasons* (Darwiche and Hirth 2020), which aim to explain a decision $\mathcal{M}(\boldsymbol{x}) = 1$ by presenting a subset S of the features of the input $\boldsymbol{x}$ that implies $\mathcal{M}(\boldsymbol{z}) = 1$ for any $\boldsymbol{z}$ that agrees with $\boldsymbol{x}$ on S. In the language of theoretical computer science, these correspond to *certificates* for $\mathcal{M}(\boldsymbol{x})$.

Example 1. *Consider a binary classifier $\mathcal{M}$ defined as*

$$\mathcal{M}(\boldsymbol{x}) = (x_1 \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_1}) \wedge (x_4 \vee x_3)\,,$$

and the input instance $\boldsymbol{x} = (1, 1, 0, 1)$. We can say that $\mathcal{M}(\boldsymbol{x}) = 1$ "because" $x_1 = 1, x_2 = 1$, and $x_4 = 1$, as they are sufficient to determine the value of $\mathcal{M}(\boldsymbol{x})$ regardless of x_3.

Let us start formalizing the framework for our work. First, we consider binary boolean models $\mathcal{M} \colon \{0,1\}^d \to \{0,1\}$. Despite our domain being binary, we will need a third value, $\perp$, to denote *"unknown"* values. For example, we may represent a person who *does* have a car, *does not* have a house, and for whom we do not know if they have a pet or not, as $(1, 0, \perp)$. We say elements of $\{0,1,\perp\}^d$ are *partial instances*, while elements of $\{0,1\}^d$ are simply *instances*. To illustrate, in Example 1 we used the partial instance $\boldsymbol{y} = (1, 1, \perp, 1)$ to explain $\mathcal{M}(\boldsymbol{x}) = 1$. We use the notation $\boldsymbol{y} \subseteq \boldsymbol{x}$ to denote that the (partial) instance $\boldsymbol{x}$ *"fills in"* values of the partial instance $\boldsymbol{y}$; more formally, we use $\boldsymbol{y} \subseteq \boldsymbol{x}$ to mean that $y_i = \perp \vee y_i = x_i$ for every $i \in [d]$. Finally, for any partial instance $\boldsymbol{y}$ we denote by $\mathrm{Comp}(\boldsymbol{y})$ the set of instances $\boldsymbol{x}$ such that $\boldsymbol{y} \subseteq \boldsymbol{x}$, thinking of $\mathrm{Comp}(\boldsymbol{y})$ as the set of *completions* of $\boldsymbol{y}$. One can define sufficient reasons as follows with this notation.

Definition 1 (Sufficient Reason (Darwiche and Hirth 2020))**.** *We say $\boldsymbol{y}$ is a* sufficient reason *for $\boldsymbol{x}$ if for any completion $\boldsymbol{z} \in \mathrm{Comp}(\boldsymbol{y})$ it holds that $\mathcal{M}(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{z})$.*

A crucial factor for the helpfulness of sufficient reasons as explanations is their size; even though $\boldsymbol{x}$ is always a sufficient reason for its own classification, we long for explanations that are much smaller than $\boldsymbol{x}$ itself. Miller (1956), for instance, goes on to say that explanations consisting of more than 9 features are probably too large for human stakeholders. In general, empirical research suggests that explanations ought to be small (Narayanan et al. 2018; Lage et al. 2019). There are several ways of formalizing the succinctness we desire for sufficient reasons:

- **(Minimum Size)** For a sufficient reason $\boldsymbol{y}$, we define its *explanation size* $|\boldsymbol{y}|_e$ as the number of defined features in $\boldsymbol{y}$, or equivalently, $|\boldsymbol{y}|_e := d - |\boldsymbol{y}|_\perp$, where $|\boldsymbol{y}|_\perp$ is the number of features of $\boldsymbol{y}$ taking $\perp$. See e.g., (Barceló et al. 2020).[1]

*The authors opted for randomized author ordering instead of alphabetical.

[1]When talking about a partial instance $\boldsymbol{y}$, we will use the "size" of $\boldsymbol{y}$ to mean $|\boldsymbol{y}|_e$.

- **(Subset minimality)** We say a sufficient reason $\boldsymbol{y}$ for a pair $(\mathcal{M}, \boldsymbol{x})$ is *minimal* if there is no other sufficient reason $\boldsymbol{y}'$ for $(\mathcal{M}, \boldsymbol{x})$ such that $\boldsymbol{y}' \subsetneq \boldsymbol{y}$. In fact, the original definition of sufficient reasons of Darwiche and Hirth (2020) includes minimality as a requirement, and so is the case under the *"abductive explanation"* naming (Ignatiev et al. 2021).
- **(Relative to average explanation)** Blanc, Lange, and Tan (2021) compute explanations that are small relative to the *"certificate complexity"* of the classifier $\mathcal{M}$, meaning the average size of the minimum sufficient reason where the average is taken over all possible instances $\boldsymbol{x}$.

Nevertheless, there is a path toward even smaller explanations: *probabilistic* sufficient reasons (Wäldchen et al. 2021; Izza et al. 2023a). As will be shown in Example 2, and is noted as a remark by Blanc, Lange, and Tan (2021), these can be significantly smaller than minimum size sufficient reasons.

2 Probabilistic Sufficient Reasons

The main idea of probabilistic sufficient reasons is to relax the condition *"all completions of the explanation $\boldsymbol{y}$ have the same class as $\boldsymbol{x}$"* to *"a random completion of $\boldsymbol{y}$ has the same class as $\boldsymbol{x}$ with high probability"*.

Let us use notation $\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y})$ to denote that $\boldsymbol{z}$ is a completion of $\boldsymbol{y}$ drawn uniformly at random. With this notation we can define δ-sufficient reasons:[2]

Definition 2 ((Wäldchen et al. 2021))**.** *For any $\delta \in [0, 1]$, a δ-sufficient reason (δ-SR) for an instance $\boldsymbol{x}$, is a partial instance $\boldsymbol{y} \subseteq \boldsymbol{x}$ such that*

$$\Pr_{\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y})} \left[\mathcal{M}(\boldsymbol{z}) = \mathcal{M}(\boldsymbol{x}) \right] \geqslant \delta.$$

Naturally, a minimum δ-SR is a δ-SR of minimum size. Note immediately that Definition 2 and Definition 1 coincide when $\delta = 1$.

2.1 The size of δ-SRs

Interestingly, even a 0.999999-SR can be arbitrarily smaller, in terms of defined features, than the smallest sufficient reason (i.e., 1-SR) for a pair $(\mathcal{M}, \boldsymbol{x})$, even when $\mathcal{M}$ is a linear model, as we will illustrate in Example 2. Before providing the example, let us define linear models.

Definition 3. *A (binary) linear model $\mathcal{L}$ of dimension d is a pair $(\boldsymbol{w}, t)$, where $\boldsymbol{w} \in \mathbb{Q}^d$ and $t \in \mathbb{Q}$. Its classification over an instance $\boldsymbol{x}$ is defined simply as*

$$\mathcal{L}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \cdot \boldsymbol{w} \geqslant t \\ 0 & \text{otherwise.} \end{cases}$$

Example 2. *Consider a linear model $\mathcal{L}$ of dimension $d = 1000$ with parameters $t = 1250$ and*

$$\boldsymbol{w} = (1000,\ 1,\ 1,\ 1,\ 1,\ \ldots,\ 1).$$

Let the instance $\boldsymbol{x}$ be $(1,\ 1,\ 1,\ 1,\ 1,\ \ldots,\ 1)$, so that clearly $\mathcal{L}(\boldsymbol{x}) = 1$. One can easily see that any 1-SR for $\boldsymbol{x}$ under $\mathcal{L}$ has size 251, as it must include the first feature and any 250 other features. However, if we consider $\boldsymbol{y} = (1,\ \perp,\ \perp,\ \perp,\ \ldots,\ \perp)$, then a simple application of the Chernoff-Hoeffding concentration bound (in the appendix for the completeness) gives that

$$\Pr_{\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y})} \Big[\mathcal{L}(\boldsymbol{z}) = 1 \Big] \geqslant 0.999999.$$

This suggests that we might say $\mathcal{L}(\boldsymbol{x}) = 1$ "because" $x_1 = 1$; formally, $\boldsymbol{y}$ is a 0.999999-SR, and 251 times smaller than any 1-SR for $\mathcal{L}(\boldsymbol{x})$.

We generalize this example as follows. Let $\textsc{Min}(\mathcal{M}, \boldsymbol{x}, \delta)$ denote the size of the smallest δ-SR for $(\mathcal{M}, \boldsymbol{x})$. Then, we have

Proposition 1. *For any $\delta \in (0, 1)$, $\gamma > 0$, and any $\varepsilon > 0$ such that $\delta + \varepsilon \leqslant 1$, there are pairs $(\mathcal{L}, \boldsymbol{x})$ where $\mathcal{L}$ is a linear model of dimension d, and $\boldsymbol{x}$ an instance of dimension d, such that*

$$\frac{\textsc{Min}(\mathcal{L}, \boldsymbol{x}, \delta + \varepsilon)}{\textsc{Min}(\mathcal{L}, \boldsymbol{x}, \delta)} = \Omega\left(d^{\frac{1}{2} - \gamma} \right).$$

Proposition 1 showcases a key subtlety of δ-SRs: a slight change in δ might lead to large changes on the smallest explanation size.

3 Approximating δ-Sufficient Reasons

Unfortunately, computing small δ-SRs is computationally challenging, even when attempting to find approximate solutions. Let us contextualize our main result by summarizing first what is known about the complexity of computing δ-SRs and their deterministic predecessors, 1-SRs.

Barceló et al. (2020) showed that computing a minimum 1-SR is Σ_2^p-hard for neural networks, NP-hard for decision trees, and polynomial-time solvable for linear models. Then, Wäldchen et al. (2021, Theorem 2.4) showed that computing minimum δ-SRs for neural networks is hard for $\mathrm{NP}^{\mathrm{PP}}$, and Arenas et al. (2022) proved that even for the restricted class of decision trees, which are usually considered interpretable, minimum δ-SRs cannot be computed in polynomial time unless $\mathrm{P} = \mathrm{NP}$ (and neither can subset-minimal δ-SRs for $\delta < 1$, in contrast to the $\delta = 1$ setting which is in P (Izza, Ignatiev, and Marques-Silva 2020; Subercaseaux 2020)). For linear models, even computing the value

$$\Pr_{\boldsymbol{z} \sim \mathbf{U}\boldsymbol{y}} \left[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x}) \right]$$

exactly is $\#\mathrm{P}$-hard (Barceló et al. 2020), from where the following is easy to show.[3]

Proposition 2. *Given a linear model $\mathcal{L}$, an instance $\boldsymbol{x}$, and $\delta \in [0, 1]$, the value $\textsc{Min}(\mathcal{L}, \boldsymbol{x}, \delta)$ cannot be computed in polynomial time unless $\mathrm{FP} = \#\mathrm{P}$.*

[2] Also known as δ-relevant sets (Izza et al. 2021; Wäldchen et al. 2021).

[3] Izza et al. (2023b) already made a more general observation of this form, but did not provide a hardness result for linear models.

Furthermore, the situation does not improve if we aim to efficiently approximate the value $\textsc{Min}(\mathcal{M}, \boldsymbol{x}, \delta)$. Wäldchen et al. (2021, Theorem 2.5) studied general classifiers (e.g., neural networks) and showed that no algorithm can achieve an approximation factor of $d^{1-\alpha}$ for this problem, where d is the dimension of the classifier and $\alpha > 0$, unless $\mathrm{P} = \mathrm{NP}$. Kozachinskiy (2023) proved that this approximation task is also hard for decision trees.

However, these hardness results do not preclude the existence of efficient algorithms for computing or approximating δ-SR for linear models. Hence, the goal of this section is to explore these questions for such models, given their practical importance.

3.1 A Simple Relaxation: (δ, ε)-min-SR

In light of the hardness results for δ-SRs, it is natural to consider a further relaxation that would allow for tractability. Consider for instance a customer of a bank who wants a 0.95-SR for why their application for a loan was rejected. Such an explanation would consist of a small number of features of their application profile that are relevant to the decision since 95% of applicants with such a profile would also get rejected. We expect that, in such a scenario, the user would not particularly care if the explanation she obtains holds for 95% of potential applicants or for 94.9997% of them. In other words, the value of δ is chosen in a trade-off between the size of the explanation and the desired level of confidence or "explanation power". We posit that in such a trade-off, the user is more sensitive to increases in the explanation size than they are to a minor perturbation in δ, the probability guarantee. As we showed in Proposition 1, by changing δ very slightly, the size of the best explanation can change significantly. This motivates the following definition:

Definition 4 ((δ, ε)-min-SR)**.** *Given a model $\mathcal{M}$, an instance $\boldsymbol{x}$, and values $\delta, \varepsilon \in (0, 1)$, we say a partial instance $\boldsymbol{y}$ of size is a (δ, ε)-min-SR if there exists a value $\delta^\star \in [\delta - \varepsilon, \delta + \varepsilon]$ such that $\boldsymbol{y}$ is a minimum $\delta^\star$-SR for $\boldsymbol{x}$ under $\mathcal{M}$.*

Note that, even though the guarantee of a (δ, ε)-min-SR is symmetric around δ, our definition is such that the ability of efficiently computing (δ, ε)-min-SRs is enough for the following two tasks:

1. A user wants an explanation as small as possible and of probability "close" to δ. Then, by computing a $(\delta - \varepsilon/2, \varepsilon/2)$-min-SR, they obtain an explanation whose probability guarantee is at most ε away from δ, and is no larger in size than the minimum δ-SR.
2. The owner of the model wants to offer a δ-SR that is as small as possible to a customer, and they want to be strict on the δ part, since offering a $(\delta - \varepsilon)$-SR would be misleading and could lead to legal issues. Then, by computing a $(\delta + \varepsilon/2, \varepsilon/2)$-min-SR, they can guarantee that the explanation is at least δ-SR, while still being likely much smaller than a minimum 1-SR.

The inapproximability result of Kozachinskiy (2023) can be translated to the (δ, ε)-min-SR problem as follows:

Theorem 1 (Kozachinskiy (2023), Theorem 1)**.** *Unless SAT can be solved in quasi-polynomial times, one cannot compute a (δ, ε)-min-SR for decision trees in polynomial time, and furthermore, any polynomial-time algorithm that guarantees to provide a δ'-SR for some $\delta' \in [\delta - \varepsilon, \delta + \varepsilon]$ will produce explanations that are up to $\Omega(d^{1-\alpha})$ times larger than any (δ, ε)-min-SR, for any $\alpha > 0$.*

Note that this hardness result for decision trees implies in turn hardness for neural networks by using standard compilation techniques (Barceló et al. 2020). Our main result is that, for linear models, we can efficiently compute (δ, ε)-min-SRs, making them the first class of models for which we have such a positive result. To state our runtime more cleanly, we use the standard notation $\widetilde{O}(f)$ to mean $O(f \cdot \log(f)^c)$ for some positive constant $c \in \mathbb{R}$.

Theorem 2. *Given a linear model $\mathcal{L}$ and an input $\boldsymbol{x}$, we can compute a (δ, ε)-min-SR successfully with probability $1 - \gamma$ in time $\widetilde{O}\left(\frac{d}{\varepsilon^2 \gamma^2}\right)$; that is, polynomial in d, $1/\varepsilon$, and $1/\gamma$.*

We remark that previous approaches for computing approximate probabilistic explanations lacked theoretical guarantees on the size of the explanations produced (Izza et al. 2023b, 2021; Izza, Meel, and Marques-Silva 2024).

In order to prove Theorem 2 we will need two main ideas: first, the fact that we can estimate the probabilities of models accepting a partial instance through sampling (which is already present in the work of Izza, Meel, and Marques-Silva (2024)), and second, that under the uniform distribution it is easy to decide which features ought to be part of small explanations over linear models.

3.2 Estimating the Probability of Acceptance

The hardness of computing $\Pr_{\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y})}[\mathcal{M}(\boldsymbol{z}) = 1]$ is about computing it to arbitrarily high precision, i.e., with an additive error within $O(2^{-d})$. However, computing a less precise estimation of $\Pr_{\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y})}[\mathcal{M}(\boldsymbol{z}) = 1]$ is simple, as the next fact (which is a direct consequence of Hoeffding's inequality) states.

Fact 1. *Let f be an arbitrary boolean function on n variables. Let M be any positive integer, and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ be M uniformly random samples from $\{0, 1\}^n$. Then*

$$\widehat{\mu}(M) := \frac{\sum_{i=1}^{M} [f(\boldsymbol{x}_i) = 1]}{M}$$

is an unbiased estimator for

$$\mu := \Pr_{\boldsymbol{x} \in \{0,1\}^n} [f(\boldsymbol{x}) = 1],$$

and $\Pr[|\widehat{\mu}(M) - \mu| \leqslant t] \geqslant 1 - 2\exp(-2t^2 M)$, which is at least $1 - \gamma$ for $M = \frac{1}{2t^2} \log(2/\gamma)$.

As a consequence of the previous idea, although a minimum δ-SR might be hard to compute, this crucially depends on the value of δ. In order to deal with this, our algorithm will sample a value $\delta^\star$ uniformly at random from $[\delta - \varepsilon, \delta + \varepsilon]$, and then compute a minimum $\delta^\star$-SR. Intuitively, the idea is that as $\delta^\star$ is chosen at random, it will be unlikely that a value that makes the computation hard is chosen.

Before proving Theorem 2, we need to prove a lemma concerning the easiness of selecting the features of the desired explanation.

3.3 Feature Selection

Even if we were granted an oracle computing the probabilities $\Pr_{\boldsymbol{z}\in\mathbf{D}(\boldsymbol{y})}[\mathcal{M}(\boldsymbol{z}) = 1]$, that would not be necessarily enough to efficiently compute a minimum δ-SR. Indeed, for decision trees, the counting problem can be easily solved in polynomial time (Barceló et al. 2020), and yet the computation of δ-SRs of minimum size is hard, even to approximate (Arenas et al. 2022; Kozachinskiy 2023). Intuitively, the problem for decision trees is that, even if we were told that the minimum δ-SR has exactly k features, it is not obvious how to search for it better than enumerating all $\binom{d}{k}$ subsets. The case of linear models, however, is different, at least under the uniform distribution. In this case, every feature i that is not part of the explanation will take value 0 or 1 independently with probability $1/2$, and contribute to the classification according to its weight w_i. In other words, we can sort the features according to their weights (with some care about signs), and select them greedily to build a small δ-SR. A proof for the deterministic case ($\delta = 1$) was already given in (Barceló et al. 2020) and sketched earlier on by (Marques-Silva et al. 2020a).

Definition 5. *Given a linear model $\mathcal{L} = (\boldsymbol{w}, t)$, and an instance $\boldsymbol{x}$, both having dimension d, we define the* score *of feature $i \in [d]$ as*

$$s_i := w_i \cdot (2x_i - 1) \cdot (2\mathcal{L}(\boldsymbol{x}) - 1).$$

In other words, the sign of s_i is $+1$ if the feature is "helping" the classification, and -1 if it is "hurting" it. The magnitude of s_i is proportional to the weight of the feature i. Changing the value of feature i in an instance $\boldsymbol{x}$ would decrease $\boldsymbol{w} \cdot \boldsymbol{x}$ by s_i if $\mathcal{L}(\boldsymbol{x}) = 1$, and increase it by s_i if $\mathcal{L}(\boldsymbol{x}) = 0$. For the uniform distribution (or more generally, any distribution in which all features are Bernoulli variables with the same parameter), we can prove the following lemma that basically states that, for linear models it is good to choose features greedily according to their score.

Lemma 1. *Given a linear model $\mathcal{L}$, and an instance $\boldsymbol{x}$, if $\boldsymbol{y}^{(0)}, \ldots, \boldsymbol{y}^{(d)}$ are the partial instances of $\boldsymbol{x}$ such that $\boldsymbol{y}^{(k)} \subseteq \boldsymbol{x}$ is defined only in the top k features of maximum score, then*

$$\Pr_{\boldsymbol{z}\sim\mathbf{U}(\boldsymbol{y}^{(k+1)})}[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x})] \geqslant \Pr_{\boldsymbol{z}\sim\mathbf{U}(\boldsymbol{y}^{(k)})}[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x})]$$

for all $0 \in \{1, \ldots, d-1\}$, and naturally,

$$\Pr_{\boldsymbol{z}\sim\mathbf{U}(\boldsymbol{y}^{(d)})}[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x})] = 1.$$

Moreover, $\textsc{Min}(\mathcal{L}, \boldsymbol{x}, \delta) = k$ if and only if $\boldsymbol{y}^{(k)}$ is a δ-SR for $\boldsymbol{x}$, and either $k = 0$ or $\boldsymbol{y}^{(k-1)}$ is not a δ-SR for $\boldsymbol{x}$.

Even though a proof of Lemma 1 is presented in the supplementary material, let us provide a self-contained example to help convince a reader of the veracity of the lemma.

Partial instance	Features included	Probability
$(1, \perp, 0, \perp, \perp)$	$\{1, 3\}$	$7/8$
$(1, \perp, \perp, 1, \perp)$	$\{1, 4\}$	$5/8$
$(\perp, \perp, 0, 1, \perp)$	$\{3, 4\}$	$1/2$
$(\perp, \perp, 0, \perp, 1)$	$\{3, 5\}$	$3/8$
$(\perp, 0, 0, \perp, \perp)$	$\{2, 3\}$	$3/8$
$(1, \perp, \perp, \perp, 1)$	$\{1, 5\}$	$3/8$
$(1, 0, \perp, \perp, \perp)$	$\{1, 2\}$	$3/8$
$(\perp, \perp, \perp, 1, 1)$	$\{4, 5\}$	$1/4$
$(\perp, 0, \perp, 1, \perp)$	$\{2, 4\}$	$1/4$
$(\perp, 0, \perp, \perp, 1)$	$\{2, 5\}$	$1/8$

Table 1: Table of probabilities associated to Example 3.

Example 3. *Consider an instance $\boldsymbol{x} = (1, 0, 0, 1, 1)$ and the linear model $\mathcal{L}$ be defined by*

$$\boldsymbol{w} = (5, 1, -3, 2, -1) \quad ; \quad t = 5.$$

It is easy to check that $\boldsymbol{w} \cdot \boldsymbol{x} = 6$, and thus $\mathcal{L}(\boldsymbol{x}) = 1$. The feature scores, according to Definition 5, are:

$$s_1 = 5,\ s_2 = -1,\ s_3 = 3,\ s_4 = 2,\ s_5 = -1.$$

For the first part, the main idea is that a positive score s_i means that the feature is helping the classification (i.e., adding it to a partial instance does not decrease its probability guarantee), while a negative score means that the feature is hurting the classification (i.e., adding it to a partial instance does not increase its probability guarantee). Because the partial instances

$$\boldsymbol{y}^{(0)} \subseteq \boldsymbol{y}^{(1)} \subseteq \cdots \boldsymbol{y}^{(d)}$$

are obtained by adding a single feature at a time, and thus features are added in decreasing order of their scores, then this procedure will have two phases: (i) First, it will add features with a positive score, which raise or maintain the probability of the classification being the same as $\boldsymbol{x}$, as the lemma says, and then (ii) it will start adding features with a negative score, which would seem to contradict the lemma, but it turns out that at that point the partial instance $\boldsymbol{y}^{(k)}$ would have probability guarantee 1; this is because $\boldsymbol{y}^{(d)} = \boldsymbol{x}$, which trivially has probability guarantee 1. Table 2 presents the probabilities associated to the partial instances $\boldsymbol{y}^{(0)}, \ldots, \boldsymbol{y}^{(d)}$.

For the second part, consider the partial instances $\boldsymbol{y}^\star = (\perp, 0, 0, 1, 1)$ and $\boldsymbol{y}^\dagger = (1, \perp, 0, 1, 1)$. The instance $\boldsymbol{x}$ is a completion of both $\boldsymbol{y}^\star$ and $\boldsymbol{y}^\dagger$, but $\boldsymbol{y}^\star$ also has completion $\boldsymbol{x}^\star = (0, 0, 0, 1, 1)$, whereas $\boldsymbol{y}^\dagger$ has also completion $\boldsymbol{x}^\dagger = (1, 1, 0, 1, 1)$. Note that $\boldsymbol{w} \cdot \boldsymbol{x}^\star = 1 = \boldsymbol{w} \cdot \boldsymbol{x} - s_1$, whereas $\boldsymbol{w} \cdot \boldsymbol{x}^\dagger = 6 = \boldsymbol{w} \cdot \boldsymbol{x} - s_2$. Intuitively, this means that it is better to keep feature 1 as part of the explanation, but not feature 2. If we want an explanation with only two features, we should choose feature 1 and feature 3, as they have the highest scores. Indeed, Table 1 presents the probabilities to all possible explanations of size 2.

With Lemma 1 in hand, we can proceed to prove Theorem 2.

Partial instance	Features	Probability	Score
$\boldsymbol{y}^{(0)}$	$(\perp, \perp, \perp, \perp, \perp)$	$1/4$	-
$\boldsymbol{y}^{(1)}$	$(1, \perp, \perp, \perp, \perp)$	$1/2$	5
$\boldsymbol{y}^{(2)}$	$(1, \perp, 0, \perp, \perp)$	$7/8$	3
$\boldsymbol{y}^{(3)}$	$(1, \perp, 0, 1, \perp)$	$1/1$	2
$\boldsymbol{y}^{(4)}$	$(1, 0, 0, 1, \perp)$	$1/1$	-1
$\boldsymbol{y}^{(5)}$	$(1, 0, 0, 1, 1)$	$1/1$	-1

Table 2: Table of probabilities associated to Example 3. The last column denotes the score of the feature added to the partial instance in that row with respect to the previous row.

Proof of Theorem 2. We use Algorithm 1. Let us define the partial instances $\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)} \ldots, \boldsymbol{y}^{(d)}$ so that $\boldsymbol{y}^{(k)} \subseteq \boldsymbol{x}$ is the partial instance defined only in the k features with maximum score (line 6). We then define a sequence of values v_k as

$$v_k := \Pr_{\boldsymbol{z} \sim \mathbf{U}(\boldsymbol{y}^{(k)})}[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x})],$$

and note that due to Lemma 1, the sequence $v_0, \ldots, v_d$ is non-decreasing. Let $M = \frac{\log^2 d}{2\varepsilon^2\gamma^2} \log(2 \log d/\gamma)$, as in line 8, and let us define random variables $\widetilde{v_k}$ as follows: if Algorithm 1 enters line 13 with $m = k$, then $\widetilde{v_k}$ is the output of Algorithm 2 (i.e., $\widehat{v}_k(M)$), and otherwise $\widetilde{v_k} = v_k$. We use binary search (lines 10-19), to find $k^\star$, the smallest k such that $\widetilde{v_k} \geqslant \delta^\star$, and our goal is to show that with good probability $k^\star$ is also the smallest k such that $v_k \geqslant \delta^\star$, which would imply the correctness of the algorithm by Lemma 1. Note, however, that even though the sequence $v_0, \ldots, v_d$ is non-decreasing (Lemma 1), the estimated values $\widehat{v_k}$ are not necessarily so. Let S be a random variable corresponding to the set of values k such that Algorithm 1 enters line 13 with $m = k$, and note that if for every k in S it happens that the events

$$A_k := (v_k \geqslant \delta^\star) \text{ and } B_k := (\widetilde{v_k} \geqslant \delta^\star)$$

are equivalent (i.e., either both occur or neither occurs), then the algorithm will succeed, as that would indeed imply that $k^\star$ is the smallest k such that $v_k \geqslant \delta^\star$.

Then, for $k \in [d]$, define events E_k and F_k as follows:

$$E_k := |\delta^\star - v_k| \geqslant \frac{\varepsilon\gamma}{\log d},$$
$$F_k := |\widetilde{v_k} - v_k| \leqslant \frac{\varepsilon\gamma}{\log d}.$$

We claim that if both E_k and F_k hold for some k, then A_k and B_k are equivalent events for that k. Indeed,

$$\begin{aligned} A_k &\iff v_k \geqslant \delta^\star \\ &\iff v_k \geqslant \delta^\star + \frac{\varepsilon\gamma}{\log d} && \text{(by } E_k\text{)} \\ &\iff v_k - \frac{\varepsilon\gamma}{\log d} \geqslant \delta^\star \\ &\iff \widetilde{v_k} \geqslant \delta^\star && \text{(by } F_k\text{)} \\ &\iff B_k. \end{aligned}$$

Algorithm 1: LinearMonteCarloExplainer

Input: Linear model $\mathcal{L}$, instance $\boldsymbol{x}$, parameters $\delta \in (0, 1)$
Parameter: $\varepsilon \in (0, 1)$, $\gamma \in (0, 1)$
Output: A value $\delta^\star \in [\delta - \varepsilon, \delta + \varepsilon]$ together with a minimum $\delta^\star$-SR explanation for $\boldsymbol{x}$.

1: $\delta^\star \leftarrow$ uniformly random sample from $[\delta - \varepsilon, \delta + \varepsilon]$
2: **for** $i \in \{1, \ldots, d\}$ **do**
3: $\quad s_i \leftarrow w_i \cdot (2x_i - 1) \cdot (2\mathcal{L}(\boldsymbol{x}) - 1)$
4: **end for**
5: **for** $k \in \{0, 1 \ldots, d\}$ **do**
6: $\quad$ Let $\boldsymbol{y}^{(k)} \subseteq \boldsymbol{x}$ be the partial instance defined only in the top k features with maximum score s_i.
7: **end for**
8: $M \leftarrow (\log^2 d)/(2\varepsilon^2\gamma^2) \log(2 \log d/\gamma)$
9: LB $\leftarrow 0$, UB $\leftarrow d$, and STEPS $\leftarrow 0$
10: **while** LB $\neq$ UB and STEPS $\leqslant \log d$ **do**
11: $\quad$ STEPS $\leftarrow$ STEPS $+ 1$
12: $\quad m \leftarrow (\text{LB} + \text{UB})/2$
13: $\quad \widehat{v}_m \leftarrow$ MONTECARLOESTIMATION$(\mathcal{L}, \boldsymbol{y}^{(m)}, \boldsymbol{x}, M)$
14: $\quad$ **if** $\widehat{v}_m \geqslant \delta^\star$ **then**
15: $\quad\quad$ UB $\leftarrow m$
16: $\quad$ **else**
17: $\quad\quad$ LB $\leftarrow (m + 1)$
18: $\quad$ **end if**
19: **end while**
20: $k^\star \leftarrow$ LB (or equivalently, UB)
21: **return** $(\delta^\star, \boldsymbol{y}^{(k^\star)})$

Thus, if we show that E_k and F_k hold with good probability for every $k \in S$, we can conclude the theorem. Notice first that, because of the condition on the variable STEPS (lines 10, 11) we have $|S| \leqslant \log d$, allowing us to do a binary search in case the desired events E_k and F_k hold, and preventing the algorithm from looping otherwise; this way the runtime is controlled not only on expectation but deterministically. [4] Then, note that for any k we have[5]

$$\begin{aligned} \Pr\left[\overline{F_k}\right] \leqslant \Pr\left[\overline{F_k} \mid k \in S\right] &= \Pr\left[|\widehat{v_k}(M) - v_k| > \frac{\varepsilon\gamma}{\log d}\right] \\ &\leqslant \frac{\gamma}{\log d}. \qquad \text{(by Fact 1)} \end{aligned}$$

Because S itself is a random variable, whose size is also a random variable, we need to be careful before applying a union bound or any related tricks. Let us refer to the elements of S as $\{s_1, \ldots, s_\ell\}$, and let us denote $F(i)$ for $i \in [\log d]$ to the event F_{s_i} if $i \leqslant \ell$, and to the sample space Ω (i.e., the event that always happens) otherwise.[6] Then, we

[4] For simplicity, we will say $|S| \leqslant \log d$, even though the exact bound for a binary search is $|S| \leqslant \lfloor \log d + 1 \rfloor$.

[5] For any event $E \subseteq \Omega$ in a probability space Ω, we use notation $\bar{E}$ to denote its complement event $\Omega \backslash E$, which holds $\Pr[\bar{E}] = 1 - \Pr[E]$.

[6] Throughout this proof, we use notation $[\alpha]$, for $\alpha \in \mathbb{R}^{>0}$, to denote the set $\{0, 1, 2, \ldots, \lceil\alpha\rceil - 1, \lceil\alpha\rceil\}$.

Algorithm 2: MonteCarloEstimation

Input: Linear model $\mathcal{L}$, a partial instance $\boldsymbol{y}$, an instance $\boldsymbol{x}$, and a number of samples M
Output: An estimate $\widehat{v}$ of $\Pr_{\boldsymbol{z}\sim\mathbf{U}(\boldsymbol{y})}[\mathcal{L}(\boldsymbol{z}) = \mathcal{L}(\boldsymbol{x})]$.

```
1: v̂ ← 0
2: for i = 1 to M do
3:     Sample z ~ U(y)
4:     if L(z) = L(x) then
5:         v̂ ← v̂ + 1
6:     end if
7: end for
8: return v̂/M
```

claim that for any $1 \leqslant i \neq j \leqslant \lceil \log d \rceil$, we have

$$\Pr[F(i) \cap F(j)] = \Pr[F(i)] \cdot \Pr[F(j)], \tag{1}$$

as either $\max\{i, j\} \leqslant \ell$, in which case the claim holds by independence (since both events depend only on disjoint sets of independent random samples), or the claim holds trivially since $\Pr[F(i)] = 1$ for $i > \ell$. Therefore, we have

$$\begin{aligned}\Pr\left[\bigcap_{k\in S} F_k\right] &= \Pr\left[F(1) \cap F(2) \cap \cdots \cap F(\lceil \log d \rceil)\right] \\ &= \prod_{i\in[\log d]} \Pr[F(i)] \qquad \text{(by Equation (1))} \\ &\geqslant \left(1 - \frac{\gamma}{\log d}\right)^{\log d} \geqslant 1 - \gamma.\end{aligned}$$

We now argue that the event $\bigcap_{k\in S} E_k$ happens with good probability. To see that, note first that for every $k \in [d]$, line 1 implies

$$\Pr[\overline{E_k}] = \Pr\left[\delta^\star \in \left[v_k \pm \frac{\varepsilon\gamma}{\log d}\right]\right] \leqslant \frac{\frac{2\varepsilon\gamma}{\log d}}{2\varepsilon} = \frac{\gamma}{\log d}.$$

Once again, we need to be careful as the events E_k are not independent of S, nor between them this time. Using the law of total probabilities, we have

$$\begin{aligned}\Pr\left[\bigcap_{k\in S} E_k\right] &= \sum_{S'\subseteq[d]} \Pr\left[S = S' \mid \bigcap_{k\in S'} E_k\right] \Pr\left[\bigcap_{k\in S'} E_k\right] \\ &= \sum_{\substack{S'\subseteq[d] \\ |S'|\leqslant\log d}} \Pr\left[S = S' \mid \bigcap_{k\in S'} E_k\right] \Pr\left[\bigcap_{k\in S'} E_k\right],\end{aligned}$$

where we can now effectively use the union bound to say that for any fixed S' with $|S'| \leqslant \log d$ we have

$$\Pr\left[\bigcap_{k\in S'} E_k\right] \geqslant 1 - \gamma.$$

Therefore, we conclude that

$$\begin{aligned}\Pr\left[\bigcap_{k\in S} E_k\right] &= \sum_{\substack{S'\subseteq[d] \\ |S'|\leqslant\log d}} \Pr\left[S = S' \mid \bigcap_{k\in S'} E_k\right] \Pr\left[\bigcap_{k\in S'} E_k\right] \\ &\geqslant (1-\gamma) \sum_{\substack{S'\subseteq[d] \\ |S'|\leqslant\log d}} \Pr\left[S = S' \mid \bigcap_{k\in S'} E_k\right]. \qquad (\dagger)\end{aligned}$$

Recall that $\Pr[F_k | k \notin S] \geqslant \Pr[F_k | k \in S]$ for any k, from where it follows that for every set S' of size at most $\log d$ we have $\Pr\left[\bigcap_{k\in S'} F_k\right] \geqslant \Pr\left[\bigcap_{k\in S} F_k\right]$. Then, note that for any index $k \in [d]$, the event F_k is conditionally independent of all events E_j, with $j \in [d]$ given the event $k \in S$. That is, $\Pr\left[F_k | k \in S\right] = \Pr\left[F_k | E_j, k \in S\right]$ for any $j \in [d]$. We thus deduce that for any fixed S' with $|S'| \leqslant \log d$ we have

$$\begin{aligned}\Pr\left[\bigcap_{k\in S'} F_k \mid \bigcap_{k\in S'} E_k\right] &\geqslant \Pr\left[\bigcap_{k\in S'} F_k \mid \bigcap_{k\in S'} E_k \cap \{k \in S\}\right] \\ &= \Pr\left[\bigcap_{k\in S'} F_k \mid \bigcap_{k\in S'} k \in S\right] \\ &\geqslant \Pr\left[\bigcap_{k\in S} F_k\right] \geqslant (1-\gamma).\end{aligned} \tag{2}$$

Then our key observation is that there is a single value $S^\star \subseteq [d]$, with $|S^\star| \leqslant \log d$, that the binary search can take if we condition on all the events E_k and F_k happening, since in that case events A_k and B_k coincide. In other words, there exists $S^\star$, with $S^\star \subseteq [d]$ and $|S^\star| \leqslant \log d$, such that

$$\Pr\left[S = S^\star \mid \bigcap_{k\in S^\star} E_k \cap \bigcap_{k\in S^\star} F_k\right] = 1.$$

We can then argue as follows:

$$\begin{aligned}\Pr\left[\bigcap_{k\in S} E_k\right] &\geqslant (1-\gamma) \sum_{\substack{S'\subseteq[d] \\ |S'|\leqslant\log d}} \Pr\left[S = S' \mid \bigcap_{k\in S'} E_k\right] \\ &\qquad \text{(by } (\dagger)\text{)} \\ &\geqslant (1-\gamma) \Pr\left[S = S^\star \mid \bigcap_{k\in S^\star} E_k\right] \\ &\geqslant (1-\gamma) \Pr\left[S = S^\star \cap \bigcap_{k\in S^\star} F_k \mid \bigcap_{k\in S^\star} E_k\right] \\ &= (1-\gamma) \Pr\left[S = S^\star \mid \bigcap_{k\in S^\star} E_k \cap \bigcap_{k\in S^\star} F_k\right] \\ &\quad \cdot \Pr\left[\bigcap_{k\in S^\star} F_k \mid \bigcap_{k\in S^\star} E_k\right] \\ &\geqslant (1-\gamma)^2. \qquad \text{(by Equation (2))}\end{aligned}$$

Therefore, the algorithm will succeed with probability at least

$$\Pr\left[\bigcap_{k\in S} E_k\right] \cdot \Pr\left[\bigcap_{k\in S} F_k\right] \geqslant (1-\gamma)^3 \geqslant 1-3\gamma.$$

The runtime is simply $O(\log d \cdot M \cdot d)$; as (i) the binary search performs $O(\log d)$ steps; (ii) each of the binary search steps requires M samples, and (iii) each sample requires evaluating the model $\mathcal{L}$ and thus takes time $O(d)$. Naturally, running the algorithm with $\gamma' = 1/3 \cdot \gamma$ will yield a success probability of $1-\gamma$ without changing the asymptotic runtime, and thus we conclude the proof.

□

4 Locally Minimal Probabilistic Explanations

Due to the complexity of finding even subset-minimal δ-SR, Izza, Meel, and Marques-Silva (2024) have proposed to study "locally minimal" δ-SR, which are δ-SRs such that the removal of any feature from the explanation would decrease its probabilistic guarantee below δ. Interestingly, we can generalize a proof from (Arenas et al. 2022) to show that, over lineal models even in the more general case of product distributions (distributions over $\{0,1\}^d$ that are products of independent Bernoulli variables of potentially different parameters), every locally minimal δ-SR is a subset-minimal δ-SR. This allows leveraging the previous results of Izza, Meel, and Marques-Silva (2024) to subset-minimal δ-SRs in the case of linear models.

Theorem 3. *For linear models, under any product distribution, every locally minimal δ-SR is a subset-minimal δ-SR.*

Proof sketch. Define the *"locality"* gap $\textsc{lgap}(\boldsymbol{y})$ of a locally minimal δ-SR $\boldsymbol{y}$ as the smallest value g such that $|\boldsymbol{y}^\star|_\perp - |\boldsymbol{y}|_\perp = g$ for some $\boldsymbol{y}^\star \subseteq \boldsymbol{y}$ that is a δ-SR. If $g = 0$, then $\boldsymbol{y}$ is globally minimal, and we are done. If g were to be 1, then $\boldsymbol{y}$ would not be locally minimal, a contradiction. Therefore, we can safely assume $g \geqslant 2$ from now on. Let $\mathcal{L}, \boldsymbol{y}$ be such that $\boldsymbol{y}$ is locally minimal δ-SR and $\textsc{lgap}(\boldsymbol{y}) \geqslant 2$. We will find a contradiction by the following method:

- Let $\boldsymbol{y}^\star$ be the δ-SR such that $|\boldsymbol{y} \backslash \boldsymbol{y}^\star| = \textsc{lgap}(\boldsymbol{y})$.
- Every feature in $\boldsymbol{y} \backslash \boldsymbol{y}^\star$ is either "good", if its score is positive, or "bad" if its score is negative.
- Fix any feature i in $\boldsymbol{y} \backslash \boldsymbol{y}^\star$. If i is good, then $\boldsymbol{y}^\star \oplus i$, meaning the partial instance obtained by taking $\boldsymbol{y}$ and setting its i-th feature to x_i, has a probability guarantee greater or equal than that of $\boldsymbol{y}^\star$ (the proof of this fact is very similar to the proof of Lemma 1), and the gap has reduced. On the other hand, if i is bad, then $\boldsymbol{y} \ominus i$, meaning the partial instance obtained from $\boldsymbol{y}$ by setting $y_i = \perp$, has greater-equal probability than $\boldsymbol{y}$, contradicting the fact that $\boldsymbol{y}$ is locally minimal.

□

5 Conclusion and Future Work

We have proved a positive result for the case of linear models, showing that a (δ, ε)-min-SRs can be computed efficiently, and also a more abstract reason suggesting that linear models might be easier to explain than, e.g., decision trees. However, a variety of natural questions and directions of research remain open. First, even though the runtime of Theorem 2 is polynomial and only has a quasi-linear dependency on d, our future work includes lowering the dependency in $1/\varepsilon$ and $1/\gamma$; on a dataset with $d = 500$, setting $\varepsilon = 0.1$ and $\gamma = 0.01$ is already computationally expensive. We acknowledge, in terms of practical implementations and heuristics, the work of Bounia and Koriche (2023); Izza, Meel, and Marques-Silva (2024); Izza et al. (2023b).

Second, our theoretical result has some natural directions for generalization. We considered only binary features, whereas in order to offer a practically useful tool to the community, we will need to understand how to compute (approximate) probabilistic explanations for mixtures real-valued features and categorical features, for example under the "extended linear classifier" definition of Marques-Silva et al. (2020b). Another fascinating theoretical question is handling the generalization of our setting to that of product distributions (i.e., feature i takes value 1 with probability p_i and 0 otherwise) can also be solved efficiently. A straightforward extension of our techniques does not seem to work on such a generalized setting, since the *feature selection* argument of Section 3.3 no longer holds. Therefore, we believe that new techniques will be needed.

Third, it would be interesting to allow for a more declarative way of specifying the probabilistic guarantees or constraints on the explanations. While a recent line of research has studied the design of languages for defining explainability queries with a uniform algorithmic treatment (Arenas et al. 2021; Barceló, Pérez, and Subercaseaux 2020; Arenas et al. 2024), we are not aware of any work on that line that allows for probabilistic terms.

Acknowledgments

Bernardo Subercaseaux is supported by the U.S. National Science Foundation under grant DMS-2434625. Arenas is funded by ANID - Millennium Science Initiative Program - Code ICN17002.

References

Arenas, M.; Báez, D.; Barceló, P.; Pérez, J.; and Subercaseaux, B. 2021. Foundations of Symbolic Languages for Model Interpretability. In *Advances in Neural Information Processing Systems*, volume 34, 11690–11701. Curran Associates, Inc.

Arenas, M.; Barceló, P.; Romero Orth, M.; and Subercaseaux, B. 2022. On Computing Probabilistic Explanations for Decision Trees. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 28695–28707. Curran Associates, Inc.

Arenas, M.; Barceló, P.; Bustamante, D.; Caraball, J.; and Subercaseaux, B. 2024. A Uniform Language to Explain

Decision Trees. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, 60–70.

Barceló, P.; Monet, M.; Pérez, J.; and Subercaseaux, B. 2020. Model Interpretability through the lens of Computational Complexity. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15487–15498. Curran Associates, Inc.

Barceló, P.; Pérez, J.; and Subercaseaux, B. 2020. Foundations of Languages for Interpretability and Bias Detection. *AFCI workshop at NeurIPS 2020. Algorithmic Fairness through the Lens of Causality and Interpretability*.

Blanc, G.; Lange, J.; and Tan, L.-Y. 2021. Provably efficient, succinct, and precise explanations. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Bounia, L.; and Koriche, F. 2023. Approximating probabilistic explanations via supermodular minimization. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI '23. JMLR.org.

Darwiche, A.; and Hirth, A. 2020. On The Reasons Behind Decisions. (arXiv:2002.09284). ArXiv:2002.09284 [cs].

Ignatiev, A.; Narodytska, N.; Asher, N.; and Marques-Silva, J. 2021. From Contrastive to Abductive Explanations and Back Again. In Baldoni, M.; and Bandini, S., eds., *AIxIA 2020 – Advances in Artificial Intelligence*, Lecture Notes in Computer Science, 335–355. Cham: Springer International Publishing. ISBN 978-3-030-77091-4.

Izza, Y.; Huang, X.; Ignatiev, A.; Narodytska, N.; Cooper, M.; and Marques-Silva, J. 2023a. On computing probabilistic abductive explanations. *International Journal of Approximate Reasoning*, 159: 108939.

Izza, Y.; Huang, X.; Ignatiev, A.; Narodytska, N.; Cooper, M.; and Marques-Silva, J. 2023b. On Computing Probabilistic Abductive Explanations. *International Journal of Approximate Reasoning*, 159: 108939.

Izza, Y.; Ignatiev, A.; and Marques-Silva, J. 2020. On Explaining Decision Trees.

Izza, Y.; Ignatiev, A.; Narodytska, N.; Cooper, M. C.; and Marques-Silva, J. 2021. Efficient Explanations With Relevant Sets. *ArXiv*, abs/2106.00546.

Izza, Y.; Meel, K. S.; and Marques-Silva, J. 2024. Locally-Minimal Probabilistic Explanations. arXiv:2312.11831.

Kozachinskiy, A. 2023. Inapproximability of sufficient reasons for decision trees. ArXiv:2304.02781 [cs].

Lage, I.; Chen, E.; He, J.; Narayanan, M.; Kim, B.; Gershman, S.; and Doshi-Velez, F. 2019. An Evaluation of the Human-Interpretability of Explanation. ArXiv:1902.00006 [cs, stat].

Marques-Silva, J.; Gerspacher, T.; Cooper, M. C.; Ignatiev, A.; and Narodytska, N. 2020a. Explaining Naive Bayes and Other Linear Classifiers with Polynomial Time and Delay. *CoRR*, abs/2008.05803.

Marques-Silva, J.; Gerspacher, T.; Cooper, M. C.; Ignatiev, A.; and Narodytska, N. 2020b. Explaining Naive Bayes and Other Linear Classifiers with Polynomial Time and Delay. In *NeurIPS*.

Marques-Silva, J.; and Ignatiev, A. 2022. Delivering Trustworthy AI through Formal XAI. In *AAAI*.

Miller, G. A. 1956. The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63(2): 81–97.

Narayanan, M.; Chen, E.; He, J.; Kim, B.; Gershman, S.; and Doshi-Velez, F. 2018. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. ArXiv:1802.00682 [cs].

Subercaseaux, B. 2020. *Model Interpretability through the Lens of Computational Complexity*. Master's thesis, Universidad de Chile.

Wäldchen, S.; MacDonald, J.; Hauch, S.; and Kutyniok, G. 2021. The Computational Complexity of Understanding Binary Classifier Decisions. *J. Artif. Intell. Res.*, 70: 351–387.