# Towards Probabilistic Sufficient Explanations

**Eric Wang** [1] **Pasha Khosravi** [1] **Guy Van den Broeck** [1]

## Abstract

Explaining and understanding the behavior of machine learning models is an important task, and there have been many different approaches to explanation: logical reasoning tools, black-box explanations, and model-specific methods. This paper introduces *sufficient explanations* as a principled probabilistic framework for defining explanations, and discusses its relation to other methods. We introduce two kinds of sufficient explanations, provide theoretical bounds between them, and use these bounds to devise a pruning algorithm for reducing the search space for finding our explanations. We showcase our algorithm with some preliminary experiments to illustrate how sufficient explanations can provide both intuitive and principled explanations.

## 1. Introduction

Machine learning models are becoming ubiquitous, and are being used in critical and sensitive areas such as medicine, loan applications, and predicting risk assessment in courts. Hence, unexpected and faulty behaviours in machine learning models can have significant negative impact on people. As a result, there is much focus on explaining and understanding behavior of such models. Explainable AI (or XAI) is an active area of research that aims to tackle these issues.

There have been many approaches toward explaining an instance of a classification (called a local explanation) from different perspectives, including logic-based (Shih et al., 2018; Ignatiev et al., 2019a; Darwiche & Hirth, 2020) or model-agnostic approaches (Ribeiro et al., 2016; Lundberg & Lee, 2017; Ribeiro et al., 2018). Each of these methods have their pros and cons; some focus on scalability and flexibility, and some focus on providing guarantees.

[1]Department of Computer Science, University of California, Los Angeles. Correspondence to: Eric Wang <ericzxwang@ucla.edu>.

Presented at *XXAI: Extending Explainable AI Beyond Deep Models and Classifiers workshop hosted by the $37^{th}$ International Conference on Machine Learning (ICML).* 

In this work, our specific goal is to *probabilistically* explain an instance of classification. Explanations are partial examples, where we treat the features not in the explanation as missing values. We aim for our explanations to provide the following probabilistic guarantee: given only the features in the explanation, with high probability under the data distribution $\Pr(\mathbf{x})$, the classifier makes the same prediction as on the full example. We employ probabilistic reasoning tools to choose the minimal subset of features that forms the explanation, which we refer to as a *Sufficient Explanation*.

Section 2 gives a brief overview of other approaches to local explanations and discusses their pros and cons. Broadly, model-agnostic methods are more scalable and flexible but tend to be not as reliable as logic-based methods. Then, we introduce the probabilistic reasoning tools needed to define sufficient explanations: *Same-Decision Probability* (SDP), and *Expected Prediction* (EP). Additionally, we use probabilistic circuits (Choi et al., 2020) to model the probability distribution $\Pr(\mathbf{x})$ over the features.

Section 3 lays the theoretical foundation of our approach by introducing two kinds of sufficient explanations. They differ only in the probabilistic reasoning tool used to define the notion of sufficiency, that is, either SDP or EP. We discuss how these notions of sufficiency are related, and how they quantitatively bound each other, and how they can be computed in practice. Finally, we argue that probabilistic sufficiency can provide an intuitive and principled way of thinking about explanations.

Section 4 designs a search algorithm to find the most likely sufficient explanation for a given instance of classification. It uses our theoretical bounds on sufficiency to prune the search space of possible explanations. We start with the empty explanation with no observed features, and then in each iteration ask our probabilistic circuit density estimator to expand the explanation with the most likely observed features. We continue the search until we find the desired minimal sufficient explanation.

Section 5 provides some preliminary experiments to give concrete examples of explanations produced by our method and the effectiveness of our pruning algorithm. In particular we give some examples of sufficient explanations for two different classifiers, with one being a highly biased classifier.

## 2. Background and Motivation

**Notation** We use uppercase letters ($X$) for features (random variables) and lowercase letters ($x$) for their value assignments. Analogously, we denote sets of features in bold uppercase ($\mathbf{X}$) and their assignments in bold lowercase ($\mathbf{x}$). We denote the set of all possible assignments to $\mathbf{X}$ as $\mathcal{X}$. Concatenation $\mathbf{XY}$ denotes the union of disjoint sets. We will be explaining classifiers, so we use $C$ as a special random variable to denote the class variable. We focus on discrete features unless otherwise noted.

We represent a probabilistic predictor as $f : \mathcal{X} \to [0, 1]$ and its thresholded classifier as $\mathcal{C} : \mathcal{X} \to \{0, 1\}$, where $T_c$ denotes the decision threshold. Hence, $\mathcal{C}(\mathbf{x}) = [\![f(\mathbf{x}) \geq T_c]\!]$, where $[\![\Delta]\!] = 1$ if and only if $\Delta$ is true.

### 2.1. Related Work

Computing explanations of classifiers has been studied from many different perspectives, including logical reasoning, black-box methods, and model-specific approaches. Some try to explain the learned model globally, making it more interpretable (Liang & Van den Broeck, 2017; 2019), while others focus more locally on explaining its prediction for a single instance. Next, we go over some local explanation methods, discuss their pros and cons, after which we motivate how our framework might solve those issues.

**Model Agnostic Approaches** These methods treat the classifier as a black box. Given an input instance to explain, they perturb the instance in many different ways and evaluate the model on those perturbed instances. Then, they use the results from the perturbations to generate an explanation. Two popular methods under this umbrella are Lime (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017). The main difference between these methods is the heuristics used to obtain perturbed instances and how to analyze the predictions on these local perturbations. Most provide feature attributions, which are real-valued numbers assigned to each feature, to indicate their importance to the decision and in what direction.

A benefit of these methods is that they can be used to explain any model and are generally more flexible and scalable than their alternatives. On the other hand, the downsides are that they can be very sensitive to the choice of local perturbation, and might produce over-confident results (Ignatiev et al., 2019b), or be fooled by adversarial methods (Slack et al., 2020; Dimanov et al., 2020).

One of the main reasons for these downsides is that the distribution of the local perturbations tends to be different from the data distribution the classifier was originally trained on. Hence, these approaches do not benefit from the intended generalization guarantees of machine learning models. Additionally, some of the perturbations might be low probability or even impossible inputs, and we might not care about how the model behaves on those inputs.

**Logical Reasoning Approaches** These methods provide explanations with some principled guarantees by leveraging logical reasoning tools. Some approaches use knowledge compilation and tractable Boolean circuits (Shih et al., 2018; Darwiche & Hirth, 2020; Shi et al., 2020), some adopt the framework of abductive reasoning (Ignatiev et al., 2019a;b), and some tackle a specific family of models such as tree ensembles (Devos et al., 2020).

The main benefit of these approaches is that they guarantee provably correct explanations, in that they guarantee a certain prediction for all examples described by the explanation. On the other hand, one downside is that they are generally not as scalable (in the number of features) as the black-box methods. Another downside is that they need to remove the uncertainty from the classifier to be able to use logical tools and hence become more rigid. Particularly, in order to guarantee a certain outcome with certainty, it is often necessary to include almost all of the features into the explanation, making it much less informative.

Sufficient Reasons (Shih et al., 2018; Darwiche & Hirth, 2020) is one example of these methods that selects as an explanation a minimal subset of features that guarantees that, no matter what is observed for the remaining features, the decision will remain the same. As we see in Section 3.1, sufficient reasons, as well as related logical explanations, can be thought of as a deterministic special case our probabilistic sufficient explanations.

For a recent comparison of logic-based vs. model-agnostic explanation methods, we refer to Ignatiev et al. (2019b); Ignatiev (2020).

### 2.2. Motivation

We will overcome the limitations of both the model-agnostic and logic-based approaches by building local explanation methods that are aware of the distributions over features $\Pr(\mathbf{x})$. This distribution will allow us to (i) reason about the classifier's behavior on realistic input instances, and (ii) provide probabilistic guarantees on the veracity of the explanations. We thus take a principled probabilistic approach in explaining an instance of classification.

Intuitively, given an instance $\mathbf{x}$ and the classifiers outcome $c = \mathcal{C}(\mathbf{x})$, we would like to choose a subset of features $\mathbf{y} \subseteq \mathbf{x}$ as the "simplest sufficient explanation." Firstly, we want it to be sufficient, which means having some probabilistic guarantees about the outcome of the classifier when only features $\mathbf{y}$ are observed. Secondly, we want to choose the simplest possible subset for some definition of simplicity. In

this paper, we formalize two different versions of sufficient explanations and explore their relation.

There are a few other explanation methods with related goals. Notably, Anchors (Ribeiro et al., 2018) can be thought of as an empirical approximation of probabilistic sufficient explanations, as it aims to provide high-probability guarantees for the explanations based on local perturbations. Moreover, probabilistic sufficient explanations were used to explain logistic regression models w.r.t. Naive Bayes data distributions in Khosravi et al. (2019b).

### 2.3. Probabilistic Reasoning Tools

Before we formally define sufficient explanations in Section 3, we first introduce the probabilistic reasoning tools that support our method.

Probabilistic reasoning is a hard task in general, so we need to choose our probabilistic model carefully. We choose *probabilistic circuits*, which given some structural constrains enable tractable and exact computation of probabilistic reasoning queries such as marginals (Choi et al., 2020). Moreover, they do so without giving up much expressivity. Another advantage of probabilistic circuits is that we can learn their structure and parameters from data, and hence avoid the exponential worst-case behavior of other probabilistic models.

The two main probabilistic reasoning tools that we use for our explanations are *Same Decision Probability* (SDP) (Chen et al., 2012), and *Expected Prediction* (EP) (Khosravi et al., 2019b). We introduce them next, in order to define two kinds of Sufficient Explanations in Section 3, and explore their trade-offs and connections to other explanations.

First we have SDP (Chen et al., 2012), which intuitively, given some subset of observed features $\mathbf{y}$, gives us the probability that our classifier has the same output as $\mathcal{C}(\mathbf{x})$.[1]

**Definition 1** (Same Decision Probability)**.** *Given a classifier $\mathcal{C}$, a distribution $\Pr(\mathbf{X})$ over features, a partition $\mathbf{YM}$ of features $\mathbf{X}$, and an assignment $\mathbf{y}$ to $\mathbf{Y}$, the same decision probability (SDP) of $\mathbf{y}$ w.r.t. $\mathbf{x}$ is*

$$SDP_{\mathcal{C},\mathbf{x}}(\mathbf{y}) = \mathop{\mathbb{E}}_{\mathbf{m}\sim\Pr(\mathbf{M}|\mathbf{y})} [\![\mathcal{C}(\mathbf{ym}) = \mathcal{C}(\mathbf{x})]\!].$$

SDP gives the probability of the decision remaining the same had we observed all the features conditioned on observing $\mathbf{y}$. The higher the SDP the better guarantee we get for partial example $\mathbf{y}$ being classified the same way as full example $\mathbf{x}$. SDP and related notions have been successfully used in applications such as trimming Bayesian network classifiers (Choi et al., 2017), and robust feature selection (Choi & Van den Broeck, 2018). Renooij (2018) introduced various theoretical properties and bounds on the SDP.

Expected Prediction is another probabilistic reasoning task that has shown to be successful in handling missing values in classification (Khosravi et al., 2019a;b; 2020). It provides a promising alternative for SDP toward explanations. Intuitively, given some partial observation, expected prediction can be thought of as trying all the possible ways of imputing the remaining features, computing an average of all the subsequent predictions, as weighted by the probability of each imputation. More formally:

**Definition 2** (Expected Prediction)**.** *Given a probabilistic predictor $f$, a distribution $\Pr(\mathbf{X})$ over features, a partition $\mathbf{YM}$ of features $\mathbf{X}$, and an assignment $\mathbf{y}$ to $\mathbf{Y}$, the expected prediction of $f$ on $\mathbf{y}$ is*

$$\mathcal{F}_f(\mathbf{y}) = \mathop{\mathbb{E}}_{\mathbf{m}\sim\Pr(\mathbf{M}|\mathbf{y})} f(\mathbf{ym}).$$

In Section 3.3, we will show how to use the expected prediction as a lower bound on the same-decision probability.

## 3. Sufficient Explanations

To explain the decision of a classifier on an instance $\mathbf{x}$, we want to choose a minimal subset of the features that best explain the classifier's decision on this instance. Next, we introduce a probabilistic framework for sufficient explanations that provide probabilistic guarantees. We develop two kinds of sufficient explanations in Sections 3.1 and 3.2 and then discuss their relation in Section 3.3.

### 3.1. Sufficient Explanations Using SDP

In this section, we use SDP as a tool to choose a minimal subset of features as explanations so that, given only the explanation, the classifier makes the same decision with high probability. More formally:

**Definition 3** (SDP Sufficient Explanation)**.** *Let $\mathcal{C}$ be a classifier and $\mathbf{x}$ be an instance that we wish to explain. A subset $\mathbf{y}$ of $\mathbf{x}$ is called a SDP Sufficient Explanation (SDP-SE) of $\mathbf{x}$ for probability $\pi$ if*

*(i)* $SDP_{\mathcal{C},\mathbf{x}}(\mathbf{y}) \geq \pi$ *(sufficiency)*

*(ii) no subset $\mathbf{z}$ of $\mathbf{y}$ satisfies (i)* *(minimality)*

Intuitively, a SDP-SE of $\mathbf{x}$ for probability $\pi$ is a minimal subset of $\mathbf{x}$ which guarantees that, with probability at least $\pi$, the classifier would make the same decision after observing the remaining features. Hence, many logical explanations discussed in Section 2 are special cases of SDP-SE. Indeed, if we wish to logically guarantee that the classifier will always make the same decision, we can take $\pi = 1$. The

[1] SDP was originally defined for the classifier being a conditional probability test in distribution $\Pr$. Here, we slightly generalize SDP to apply to a distribution $\Pr$ with a separate classifier $\mathcal{C}$.

next lemma follows directly from the minimality property of Definition 3.

**Lemma 1.** *Given an instance* $\mathbf{x}$*, a subset of features* $\mathbf{y} \subseteq \mathbf{x}$*, and a classifier* $\mathcal{C}$*, if* $\text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}) \geq \pi$*, then there exists* $\mathbf{z} \subseteq \mathbf{y}$ *such that* $\mathbf{z}$ *is a SDP-SE of* $\mathbf{x}$ *for probability* $\pi$*.*

As a consequence of this, we have the nice property that SDP-SE's for some threshold must come from SDP-SE's for lower thresholds. That is, they can all be obtained by adding more features to SDP-SE's for lower thresholds. Formally:

**Lemma 2.** *Let* $\mathbf{x}$ *be an instance and let* $\pi_2 > \pi_1$*. If* $\mathbf{y}$ *is a SDP-SE of* $\mathbf{x}$ *for probability* $\pi_2$*, then some subset* $\mathbf{z} \subseteq \mathbf{y}$ *is a SDP-SE of* $\mathbf{x}$ *for probability* $\pi_1$*.*

*Proof.* Since $\mathbf{y}$ is a SDP-SE of $\mathbf{x}$ for probability $\pi_2$, $\text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}) \geq \pi_2 > \pi_1$. By Lemma 1, some subset $\mathbf{z} \subseteq \mathbf{y}$ is a SDP-SE of $\mathbf{x}$ for probability $\pi_1$. □

While SDP-SE is an appealing criteria to use for selecting explanations, computing the SDP exactly is known to be computationally hard. In particular, it is $PP^{PP}$-hard on Bayesian networks (Choi et al., 2012). Even for a simple Naive Bayes model and classifier, computing SDP is NP-hard (Chen et al., 2013). On the other hand, SDP-SE provides intuitive and principled explanations with guarantees for decision making.

Next, we propose a second type of sufficient explanation, based on expected predictions, which will be more tractable to compute. Afterwards, we show its relation to SDP-SE.

### 3.2. Sufficient Explanations Using Expected Prediction

Expected prediction has been shown to be useful in handling missing values for classification (Khosravi et al., 2019b;a). Here, we use expected prediction to define EP-SE, a type of probabilistic sufficient explanations. Intuitively, we want a minimal subset of features that are sufficient for their expected prediction to be higher than a given threshold.

Without loss of generality, for the remainder of paper, we assume that the classifier predicts the positive class, that is $\mathcal{C}(\mathbf{x}) = 1$, and hence $f(\mathbf{x}) \geq T_c$.

**Definition 4** (EP Sufficient Explanation)**.** *Given a probabilistic predictor* $f$ *and features* $\mathbf{x}$*, a subset* $\mathbf{y}$ *of* $\mathbf{x}$ *is called an Expected Prediction Sufficient Explanation (EP-SE) of* $\mathbf{x}$ *for threshold* $\pi \in [0, 1]$ *if*

*(i)* $\mathcal{F}_f(\mathbf{y}) \geq \pi$ *(sufficiency)*

*(ii) no subset* $\mathbf{z}$ *of* $\mathbf{y}$ *satisfies (i)* *(minimality)*

Both SDP and expected prediction are taking an expectation of the output of the classifier. The main difference is that SDP takes the expectation after thresholding. On the one hand, we can take a Bayesian interpretation of the EP-SE: it answers the question of whether a probabilistic model believes the same classification to be sufficiently likely. The SDP-SE on the other hand is a property of a deterministic model, where we do not care about the probabilistic beliefs and uncertainty about the class variable.

One advantage of expected predictions is that, unlike SDP, it can be tractably computed for many different pairs of discriminative and generative models. For example, it is known to be tractable for the following cases: (i) logistic regression using a conformant naive Bayes distribution (Khosravi et al., 2019b) (ii) decision trees w.r.t. probabilistic circuits (PCs) (Khosravi et al., 2020), (iii) discriminative circuits w.r.t. PCs (Khosravi et al., 2019a) and (iv) when both feature distribution and predictor are defined by the same PC distribution $\Pr$. In the latter case, the predictor is the conditional probability $\Pr(c \mid \mathbf{x})$, and the feature distribution is $\Pr(\mathbf{x})$. Then, expected prediction can be reduced to probabilistic marginal inference in PCs which is tractable for smooth and decomposable circuits (Choi et al., 2020).

As we see next, explanations found using expected prediction are closely related to those found using SDP.

### 3.3. Relation between SDP-SE and EP-SE

In this section, we provide theoretical bounds between SDP and expected predictions and use those bounds to relate explanations provided by SDP-SE and EP-SE. The next theorem, which is similar to Markov's inequality, shows that there is a simple relation between the expected prediction and SDP.

**Theorem 1.** *Given a probabilistic predictor* $f$*, its thresholded classifier* $\mathcal{C}$*, features* $\mathbf{x}$*, and some subset of the features* $\mathbf{y} \subseteq \mathbf{x}$*, we have:*

$$\text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}) \geq \frac{\mathcal{F}_f(\mathbf{y}) - T_c}{1 - T_c}. \tag{1}$$

*Proof.* First note $\text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}) = \Pr(f(\mathbf{ym}) \geq T_c)$ and $\mathcal{F}_f(\mathbf{y}) = \mathbb{E}[f(\mathbf{ym})]$ where $\mathbf{m} \sim \Pr(\mathbf{M}|\mathbf{y})$. Thus,

$$\begin{aligned}
\mathcal{F}_f(\mathbf{y}) &= \mathbb{E}[f(\mathbf{ym})] \\
&= \mathbb{E}[f(\mathbf{ym}) | f(\mathbf{ym}) < T_c] \Pr(f(\mathbf{ym}) < T_c) \\
&\quad + \mathbb{E}[f(\mathbf{ym}) | f(\mathbf{ym}) \geq T_c] \Pr(f(\mathbf{ym}) \geq T_c) \\
&\leq T_c(1 - \Pr(f(\mathbf{ym}) \geq T_c)) + \Pr(f(\mathbf{ym}) \geq T_c) \\
&= T_c + (1 - T_c) \Pr(f(\mathbf{ym}) \geq T_c) \\
&= T_c + (1 - T_c)\ \text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}).
\end{aligned}$$

Rearranging the terms leads to Equation 1. □

Theorem 1 provides a simple way of translating between thresholds for SDP-SEs and EP-SEs. If we want to find

SDP-SEs for probability $\pi$, we could instead try to find EP-SEs for threshold $\pi' = \pi(1 - T_c) + T_c$. By combining the previous theorem with the minimality property of sufficient explanations, we arrive at the following result, which relates explanations found using SDP and expected prediction.

**Theorem 2.** *Given the features* $\mathbf{x}$ *and* $\mathbf{y} \subseteq \mathbf{x}$ *as an EP-SE of* $\mathbf{x}$ *for threshold* $\pi$*, there exists some* $\mathbf{z} \subseteq \mathbf{y}$ *such that* $\mathbf{z}$ *is a SDP-SE of* $\mathbf{x}$ *for probability* $\pi' = \dfrac{\pi - T_c}{1 - T_c}$.

*Proof.* Since $\mathbf{y}$ is an EP-SE of $\mathbf{x}$ for threshold $\pi$, $\mathcal{F}_f(\mathbf{y}) \geq \pi$. By Theorem 1, $\text{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{y}) \geq \frac{\pi - T_c}{1 - T_c} = \pi'$. Finally, by Lemma 1, there exists some $\mathbf{z} \subseteq \mathbf{y}$ such that $\mathbf{z}$ is a SDP-SE of $\mathbf{x}$ for probability $\pi'$. □

This translation, while making computations more tractable, can produce explanations with fewer guarantees. Theorem 2 only guarantees that some subset of each EP-SE of $\mathbf{x}$ for threshold $\pi$ will be a SDP-SE of $\mathbf{x}$ for probability $\pi'$. Thus, it is possible that explanations found using this translation will be larger than need be, selecting features which are not needed to guarantee the robustness of decision. Moreover, Theorem 2 does not guarantee that, for each SDP-SE of $\mathbf{x}$ for probability $\pi'$, we will find a corresponding EP-SE of $\mathbf{x}$ for threshold $\pi$. Thus, we may miss some explanations entirely.

## 4. Finding Sufficient Explanations

In this section we describe methods only for computing EP-SEs, as expected prediction is much more tractable to compute than SDP. Since the number of explanations can be exponential, instead of computing all sufficient explanations, we find only one of them. The natural choice is to find the most likely sufficient explanation.

**Definition 5** (Most Likely EP-SE)**.** *Given a probabilistic predictor* $f$ *and features* $\mathbf{x}$*, the most likely EP-SE of* $\mathbf{x}$ *for threshold* $\pi$ *is given by*

$$\operatorname*{argmax}_{\mathbf{y} \subseteq \mathbf{x}} \Pr(\mathbf{y}) \quad \textit{s.t. } \mathbf{y} \textit{ is an EP-SE.}$$

Note that if $\mathbf{z} \subseteq \mathbf{y}$ then $\Pr(\mathbf{z}) \geq \Pr(\mathbf{y})$, so the minimality requirement of Definition 4 is automatically enforced when maximizing the likelihood. Thus, given an instance $\mathbf{x}$ and a threshold $\pi$, the task of finding the Most Likely EP-SE can be simplified to

$$\operatorname*{argmax}_{\mathbf{y} \subseteq \mathbf{x}} \Pr(\mathbf{y}) \quad \text{s.t. } \mathcal{F}_f(\mathbf{y}) \geq \pi.$$

**Algorithm 1** Find Most Likely EP-SE

**Input:** instance $\mathbf{x}$, threshold $\pi$, var-order $v$

```
maxheap ← {∅}
while maxheap is not empty do
    y ← maxheap.pop()
    if F_f(y) ≥ T then
        return y
    else if bound(y, v) ≤ π then
        continue
    else
        for z in expand(y, v) do
            maxheap.push(z)
        end for
    end if
end while
```

We solve this task by searching through the lattice consisting of all possible subsets of the instance $\mathbf{x}$. We use a simple tree search algorithm to explore the state space, considering more likely feature observations before less likely ones. To increase efficiency of search, we make sure that each state is visited at most once. In particular, we use an expand function which, given a variable order $v$, takes a subset of feature observations $\mathbf{y}$ and returns a list of features observations where each element is obtained by adding to $\mathbf{y}$ one feature appearing later in the variable ordering than the latest feature in $\mathbf{y}$. For example, for an instance $\mathbf{x} = \{x_1, x_2, x_3, x_4\}$ with the same variable ordering $v$, a call to expand($\{x_2\}, v$) would return $\{x_2, x_3\}$ and $\{x_2, x_4\}$ but not $\{x_1, x_2\}$. The following lemma makes use of the fact that we restrict the features which states can expand into.

**Lemma 3.** *Given a probabilistic predictor* $f$*, its thresholded classifier* $\mathcal{C}$*, and features* $\mathbf{x}$ *such that* $\mathcal{C}(\mathbf{x}) = c$*. Let* $\mathbf{y} \subseteq \mathbf{z} \subseteq \mathbf{x}$*. Then, for any* $\mathbf{w} \subseteq \mathbf{z} \backslash \mathbf{y}$*,* $\mathcal{F}_f(\mathbf{yw}) \leq \mathcal{F}_f(\mathbf{y}) \frac{\Pr(\mathbf{y})}{\Pr(\mathbf{z})}$.

*Proof.*

$$\begin{aligned} \mathcal{F}_f(\mathbf{yw}) = \Pr(c|\mathbf{yw}) &= \frac{\Pr(c, \mathbf{yw})}{\Pr(\mathbf{yw})} \\ &\leq \frac{\Pr(c, \mathbf{y})}{\Pr(\mathbf{z})} = \mathcal{F}_f(\mathbf{y}) \frac{\Pr(\mathbf{y})}{\Pr(\mathbf{z})}. \end{aligned}$$

□

Here $\mathbf{z}$ represents the set of features which all states expanded from $\mathbf{y}$ can contain, so $\mathbf{yw}$ then represents the possible states reachable from $\mathbf{y}$. This result provides a way to prune the search space. If, during our search, we ever reach a state $\mathbf{y}$ and are limited to selecting additional features from some set $\mathbf{z}$ where $\mathcal{F}_f(\mathbf{y}) \frac{\Pr(\mathbf{y})}{\Pr(\mathbf{z})} \leq \pi$, then we no longer need to continue searching from $\mathbf{y}$.

| | THRESHOLD ($\pi$) | MOST LIKELY EP-SE | EP | $\pi' = \frac{\pi-0.5}{1-0.5}$ | SDP | W/O BOUNDS | W/ BOUNDS |
|---|---|---|---|---|---|---|---|
| CLASSIFIER 1 | 0.8 | $1, 2, 3, 4, 6, 7, 8$ | 0.805 | 0.6 | 0.995 | 2868 | 2180 |
| | 0.825 | $1, 3, 4, 5, 6, 7, 8$ | 0.828 | 0.65 | 1.000 | 3299 | 2471 |
| | 0.85 | $1, 2, 3, 4, 5, 6, 7, 8$ | 0.857 | 0.7 | 1.000 | 3500 | 2605 |
| | 0.875 | $1, 2, 3, 4, 5, 8, 11$ | 0.879 | 0.75 | 1.000 | 4037 | 2937 |
| CLASSIFIER 2 | 0.9 | $11$ | 0.922 | 0.8 | 0.943 | 246 | 246 |
| | 0.925 | $2, 11$ | 0.949 | 0.85 | 0.965 | 333 | 333 |
| | 0.95 | $2, 10, 11$ | 0.951 | 0.9 | 0.966 | 335 | 335 |
| | 0.975 | $7, 11$ | 0.987 | 0.95 | 0.991 | 1315 | 1285 |

*Table 1.* Stats for Most Likely EP-SE of two instances given to two classifiers. The threshold ($\pi$) is the EP-SE threshold we would like to achieve. The Most Likely EP-SE shows which features were selected and the EP column gives their expected prediction. $\pi'$ is the translated SDP threshold and the SDP column gives the actual SDP. Finally, the number of states visited without and with using the bounds from Lemma 3 are given in the last two columns.

The overall search algorithm is given in Algorithm 1. The sets contained in the max-heap are ordered by their marginal probabilities. Line 3 selects the most likely features out of the ones currently being considered. Lines 4-5 check if the feature subset selected is an EP-SE. If so, since higher probability states are explored first, it will be the most likely EP-SE and is returned. Lines 6-7 use Lemma 3 to prune the search, not expanding states that cannot lead to the Most Likely EP-SE. Finally, lines 9-10 continue the search using the expand function mentioned previously.

## 5. Experiments

This section presents preliminary experiments to answer the following questions: How efficient is Algorithm 1 and how helpful are the bounds in pruning the search? How good are the explanations found and can we detect biased classifiers?

Our experiments use the adult census income data set, where the prediction task is to determine whether a given individual makes over \$50,000 per year. Features include age, sex, working class, hours worked per week, education level, nationality, etc. We binarize each feature and leave out some redundant ones, such as education number. After preprocessing, we thus obtain 12 features. A probabilistic circuit was used both to model the feature distribution and as the classifier with the classification threshold being 0.5. In this case, computing expected predictions reduces to computing marginals (Khosravi et al., 2019b).

We ran Algorithm 1 on two different classifiers, explaining their decisions on various instances using different thresholds. The data is presented in Table 1. The SDP values where calculated using brute force enumeration.

For the first classifier, we explained the decision for an individual predicted to make over \$50,000 per year. We see that for a threshold of 0.875, the features selected say that the individual is over 40 years old, is either self employed or works in the private sector or for the government, has above a high school education, is married, has an occupation that is either tech support, managerial, or a professional specialty, is female, and works under 40 hours per week. While no EP-SE's were found for higher thresholds, we see that the SDP of the EP-SE's found for lower thresholds are already very high.

For the second classifier, we explained the decision for an individual predicted to make under \$50,000 per year. We see that for a threshold of 0.975, the features selected say that the individual is not white or Asian and works less than 40 hours per week. This provides evidence that the second classifier uses race as a main factor when predicting an individual makes under \$50,000 per year.

Also, as we see from Table 1, if we do not use the bounds provided in Lemma 3, we need to explore more states for higher thresholds. This is because we stop the search the moment we find an EP-SE with the desired threshold. In fact, as we see in row 4, for a threshold of 0.875 we needed to explore nearly all states before finding the most likely EP-SE when not pruning the search. While the number of explored states is also seen in Table 1 to be increasing with higher thresholds, this is not always the case. This is because higher thresholds can allow for earlier pruning in line 6 of Algorithm 1.

## 6. Conclusion

This paper introduced sufficient explanations as a principled probabilistic approach to explaining the predictions of classifiers. Sufficient explanations provide probabilistic guarantees that a classifier would make the same prediction on all examples that match the explanation. Using theoretical properties of SDP and EP, we developed an algorithm to search for the most likely sufficient explanations.

**Acknowledgements** We thank YooJung Choi and Arthur Choi for helpful discussions. This work is partially supported by NSF grants #IIS-1943641, #IIS-1633857, #CCF-1837129, DARPA XAI grant #N66001-17-2-4032, a Sloan Fellowship, and gifts from Intel and Facebook Research.

## References


Chen, S. J., Choi, A., and Darwiche, A. The same-decision probability: A new tool for decision making. 2012.

Chen, S. J., Choi, A., and Darwiche, A. An exact algorithm for computing the same-decision probability. In *IJCAI*, 2013.

Choi, A., Xue, Y., and Darwiche, A. Same-decision probability: A confidence measure for threshold-based decisions. *International Journal of Approximate Reasoning*, 53(9):1415–1428, 2012.

Choi, Y. and Van den Broeck, G. On robust trimming of bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, July 2018.

Choi, Y., Darwiche, A., and Van den Broeck, G. Optimal feature selection for decision robustness in bayesian networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, August 2017. doi: 10.24963/ijcai.2017/215.

Choi, Y., Vergari, A., and Van den Broeck, G. Probabilistic circuits: A unifying framework for tractable probabilistic models. 2020.

Darwiche, A. and Hirth, A. On the reasons behind decisions. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, 2020.

Devos, L., Meert, W., and Davis, J. Additive tree ensembles: Reasoning about potential instances, 2020.

Dimanov, B., Bhatt, U., Jamnik, M., and Weller, A. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. In *European Conference on Artificial Intelligence*, 2020.

Ignatiev, A. Towards Trustable Explainable AI, 2020. URL https://alexeyignatiev.github.io/assets/pdf/ignatiev-ijcai20-preprint.pdf.

Ignatiev, A., Narodytska, N., and Marques-Silva, J. Abduction-based explanations for machine learning models. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jul 2019a.

Ignatiev, A., Narodytska, N., and Marques-Silva, J. On validating, repairing and refining heuristic ml explanations, 2019b.

Khosravi, P., Choi, Y., Liang, Y., Vergari, A., and Van den Broeck, G. On tractable computation of expected predictions. In *Advances in Neural Information Processing Systems*, pp. 11167–11178, 2019a.

Khosravi, P., Liang, Y., Choi, Y., and Van den Broeck, G. What to expect of classifiers? reasoning about logistic regression with missing features. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, aug 2019b.

Khosravi, P., Vergari, A., Choi, Y., Liang, Y., and Van den Broeck, G. Handling missing data in decision trees: A probabilistic approach. In *The Art of Learning with Missing Values Workshop at ICML (Artemiss)*, 2020.

Liang, Y. and Van den Broeck, G. Towards compact interpretable models: Shrinking of learned probabilistic sentential decision diagrams. In *IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2017.

Liang, Y. and Van den Broeck, G. Learning logistic circuits. In *Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*, 2019.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*. 2017.

Renooij, S. Same-decision probability: threshold robustness and application to explanation. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models (PGM)*, volume 72, pp. 368–379. PMLR, 2018.

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

Ribeiro, M. T., Singh, S., and Guestrin, C. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Shi, W., Shih, A., Darwiche, A., and Choi, A. On tractable representations of binary neural networks, 2020.

Shih, A., Choi, A., and Darwiche, A. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18. AAAI Press, 2018. ISBN 9780999241127.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020.