



HAL
open science

On the Tractability of Explaining Decisions of Classifiers

Martin Cooper, Joao Marques-Silva

► **To cite this version:**

Martin Cooper, Joao Marques-Silva. On the Tractability of Explaining Decisions of Classifiers. 27th International Conference on Principles and Practice of Constraint Programming (CP 2021), Oct 2021, Montpellier (en ligne), France. pp.21:1-21:18, 10.4230/LIPICs.CP.2021.21 . hal-03523350

HAL Id: hal-03523350

<https://hal.science/hal-03523350>

Submitted on 12 Jan 2022


HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

On the Tractability of Explaining Decisions of Classifiers

Martin C. Cooper  

IRIT, Université de Toulouse III, France

João Marques-Silva  

IRIT, CNRS, Toulouse, France

Abstract

Explaining decisions is at the heart of explainable AI. We investigate the computational complexity of providing a formally-correct and minimal explanation of a decision taken by a classifier. In the case of threshold (i.e. score-based) classifiers, we show that a complexity dichotomy follows from the complexity dichotomy for languages of cost functions. In particular, submodular classifiers allow tractable explanation of positive decisions, but not negative decisions (assuming $P \neq NP$). This is an example of the possible asymmetry between the complexity of explaining positive and negative decisions of a particular classifier. Nevertheless, there are large families of classifiers for which explaining both positive and negative decisions is tractable, such as monotone or linear classifiers. We extend tractable cases to constrained classifiers (when there are constraints on the possible input vectors) and to the search for contrastive rather than abductive explanations. Indeed, we show that tractable classes coincide for abductive and contrastive explanations in the constrained or unconstrained settings.

2012 ACM Subject Classification Theory of computation \rightarrow Machine learning theory; Theory of computation \rightarrow Problems, reductions and completeness

Keywords and phrases machine learning, tractability, explanations, weighted constraint satisfaction

Digital Object Identifier 10.4230/LIPIcs.CP.2021.21

Funding This work was supported by the AI Interdisciplinary Institute ANITI, funded by the French program “Investing for the Future – PIA3” under Grant agreement no. ANR-19-PI3A-0004.

1 Explanations of ML models

Recent work has shown that it is possible to apply formal reasoning to explainable AI, thus providing formal guarantees of correctness of explanations [39, 40, 23, 24, 14, 13, 20]¹. However, scalability quickly becomes an issue because testing the validity of an explanation may be NP-hard, or even $\#P$ -hard. As a result, more recent work focused on investigating classes of classifiers for which explanations can be found in polynomial time [2, 33, 1]. A natural question is thus which other classes of classifiers allow for formal explanations to be computed in polynomial time. This is our motivation for investigating the computational complexity of finding explanations of decisions taken by boolean classifiers. More concretely, the paper proposes conditions on the decision problems associated with classification functions, which enable finding in polynomial time a so-called abductive or contrastive explanation. Furthermore, the paper shows that several large classes of classifiers respect the proposed conditions.

We consider a boolean classification problem with two classes $\mathcal{K} = \{\oplus, \ominus\}$, defined on a set of features (or attributes) x_1, \dots, x_n , which will be represented by their indices $\mathcal{A} = \{1, \dots, n\}$. The features can either be real-valued or categorical. For real-valued features,

¹ There exist a wide range of explainable AI approaches offering no formal guarantees of correctness, e.g. [17].



© Martin C. Cooper and João Marques-Silva;

licensed under Creative Commons License CC-BY 4.0

27th International Conference on Principles and Practice of Constraint Programming (CP 2021).

Editor: Laurent D. Michel; Article No. 21; pp. 21:1–21:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

21:2 On the Tractability of Explaining Decisions of Classifiers

we have $\lambda_i \leq x_i \leq \mu_i$, where λ_i, μ_i are given lower and upper bounds. For categorical features, we have $x_i \in \{1, \dots, d_i\}$. A concrete assignment to the features referenced by \mathcal{A} is represented by an n -dimensional vector $\mathbf{a} = (a_1, \dots, a_n)$, where a_j denotes the value assigned to feature j , represented by variable x_j , such that a_j is taken from the domain of x_j . The set of all n -dimensional vectors denotes the *feature space* \mathbb{A} .

Given a classifier with features \mathcal{A} , the corresponding *decision function* is a mapping from the feature space to the set of classes, i.e. $\tau : \mathbb{A} \rightarrow \mathcal{K}$. For example, for a linear classifier, the decision function picks \oplus if $\sum_i w_i x_i > t$, and \ominus if $\sum_i w_i x_i \leq t$, for some constants w_i ($i = 1, \dots, n$) and t . Given $\mathbf{a} \in \mathbb{A}$, with $\tau(\mathbf{a}) = c$, we consider the set of feature literals of the form $(x_i = a_i)$, where x_i denotes a variable and a_i a constant.

► **Definition 1.** A *PI-explanation* [39] is a subset-minimal set $\mathcal{P} \subseteq \mathcal{A}$, denoting feature literals, i.e. feature-value pairs (taken from \mathbf{a}), such that

$$\forall (\mathbf{x} \in \mathbb{A}). \bigwedge_{j \in \mathcal{P}} (x_j = a_j) \rightarrow \tau(\mathbf{x}) = c \quad (1)$$

is true.

PI-explanations are also referred to as abductive explanations [23]. PI-explanations are analogous to prime implicants of propositional formulae: finding subset-minimal (prime) implicants rather than shortest implicants is interesting from a computational point of view since deciding the existence of an implicant of size less than k is Σ_2^P -complete [43].

► **Example 2.** We consider as a running example the case of a bank which uses a function τ to decide whether to grant a loan to a couple represented by a feature vector $\mathbf{x} = (sal_1, sal_2, age_1, age_2)$, where sal_1, sal_2 are the salaries and age_1, age_2 the ages of the two people making up the couple. Suppose that $\tau(\mathbf{x}) = \oplus$ if and only if $(\max(sal_1, sal_2) \geq sal_{\min}) \wedge (\min(age_1, age_2) \leq age_{\max})$. If \mathbf{a} corresponds to a couple who both earn more than sal_{\min} and both are younger than age_{\max} , then there are four PI-explanations for $\tau(\mathbf{a}) = \oplus$: $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$ and $\{2, 4\}$. For example, $\{1, 3\}$ means that the first and third features (sal_1 and age_1) are sufficient to explain the decision. On the other hand, if \mathbf{b} corresponds to a couple who both earn more than sal_{\min} and both are older than age_{\max} , then the only PI-explanation for $\tau(\mathbf{b}) = \ominus$ is $\{3, 4\}$ (i.e. that they are both too old).

2 Definitions

In order to study the complexity of finding explanations, and in particular to identify tractable cases, we need to place restrictions on the classifier τ . Let \mathcal{D} be a set of domains. For example, \mathcal{D} may include all intervals of the real numbers and all finite subsets of the integers. Let $\mathcal{T}^{\mathcal{D}}$ represent the family of functions $\tau : \prod_{i=1}^n D_i \rightarrow \mathcal{K}$ where each domain $D_i \in \mathcal{D}$ (i.e. the feature space \mathbb{A} is the Cartesian product of domains from \mathcal{D}). We call n the arity of τ . Recall that $\mathcal{K} = \{\ominus, \oplus\}$.

We say that $\tau : \mathbb{A} \rightarrow \mathcal{K}$ is a *\mathcal{F} -threshold classifier* if it can be represented by an objective function $f : \mathbb{A} \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$ belonging to \mathcal{F} such that an input vector $\mathbf{x} \in \mathbb{A}$ is classified as positive ($\tau(\mathbf{x}) = \oplus$) iff $f(\mathbf{x})$ is strictly greater than some threshold t , negative otherwise. Concentrating on threshold classifiers is not really a restriction, since any binary classifier $\tau : \mathbb{A} \rightarrow \{0, 1\}$ (identifying \ominus with 0 and \oplus with 1) can be viewed as a threshold classifier with $f = \tau$ and threshold $t = 0$. It is the choice of the family of functions \mathcal{F} which determines the complexity of explaining decisions.

If \mathcal{F} is the set of real-valued linear functions, then \mathcal{F} -threshold classifiers are known as linear classifiers. Similarly, we can define larger families of threshold classifiers by restricting the objective function f to be monotone or submodular. A function f is *monotone* if $\forall \mathbf{x}, \mathbf{y}, \mathbf{x} \leq \mathbf{y}$ implies $f(\mathbf{x}) \leq f(\mathbf{y})$; f is *submodular* if $\forall \mathbf{x}, \mathbf{y}, f(\min(\mathbf{x}, \mathbf{y})) + f(\max(\mathbf{x}, \mathbf{y})) \leq f(\mathbf{x}) + f(\mathbf{y})$, where \min and \max are applied componentwise [16]. All linear functions are submodular but only those linear functions whose coefficients are non-negative are monotone. Similarly, f is *antitone* if $\forall \mathbf{x}, \mathbf{y}, \mathbf{x} \leq \mathbf{y}$ implies $f(\mathbf{x}) \geq f(\mathbf{y})$; f is *supermodular* if $\forall \mathbf{x}, \mathbf{y}, f(\min(\mathbf{x}, \mathbf{y})) + f(\max(\mathbf{x}, \mathbf{y})) \geq f(\mathbf{x}) + f(\mathbf{y})$; f is *modular* if $\forall \mathbf{x}, \mathbf{y}, f(\min(\mathbf{x}, \mathbf{y})) + f(\max(\mathbf{x}, \mathbf{y})) = f(\mathbf{x}) + f(\mathbf{y})$. It is worth pointing out that all these classes of functions (linear, modular, submodular, supermodular, monotone, antitone) are closed under addition. Modular functions are exactly those functions f that can be decomposed into a sum of unary functions $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ [9]. By definition, modular functions are both submodular and supermodular and include linear functions as a special case.

Monotonicity [34] is a desirable property in applications where it is important to guarantee meritocratic fairness (do not favour a less-qualified candidate) [27]. It has been imposed even for classifiers as complex as neural networks [32].

Submodularity is a well-studied concept in Operations Research and Machine learning whose origins can be traced back to the the notion of diminishing marginal returns studied by Gaspard Monge [5]. It is well known that a submodular function over boolean domains can be minimized in polynomial time [36, 31, 6]. For example, if the objective function f is the sum of functions of pairs of variables, then minimizing f is equivalent to finding the minimum cut in a weighted graph [8]. A polynomial-time algorithm for minimizing a submodular function over any finite domains follows from the polynomial reduction to boolean domains obtained by replacing each variable x_i with domain $\{1, \dots, d\}$ by $d - 1$ boolean variables $x_{ir} = 1 \Leftrightarrow x_i \geq r$ ($r = 1, \dots, d - 1$) [9].

► **Example 3.** Consider again our example of a bank which uses a function τ to decide whether to grant a loan to a couple represented by the feature vector $\mathbf{x} = (sal_1, sal_2, age_1, age_2)$. Suppose that τ is a threshold classifier $\tau(\mathbf{x}) = \oplus \Leftrightarrow f(\mathbf{x}) > t$, where $f = \alpha f_1 + \beta f_2 + \gamma f_3$ and $f_1(\mathbf{x}) = \max(sal_1, sal_2) + \mu \min(sal_1, sal_2)$ (where $0 \leq \mu \leq 1$), and $f_2(\mathbf{x}) = 1$ iff $(\max(age_1, age_2) \geq age_{\min})$ (and $f_2(\mathbf{x}) = 0$ otherwise), and $f_3(\mathbf{x}) = 1$ iff $(\min(age_1, age_2) \leq age_{\max})$ (and $f_3(\mathbf{x}) = 0$ otherwise), where age_{\min}, age_{\max} and $\alpha, \beta, \gamma, \mu \geq 0$ are constants.

It can be verified that f_1 and f_2 are both submodular and monotone, and that f_3 is both submodular and antitone. Thus (by additivity of submodularity), f is submodular but it is neither monotone nor antitone (assuming $\alpha, \beta, \gamma > 0$). On the other hand, f is monotone if $\gamma = 0$.

We say that τ is a \mathcal{F} -*multi-threshold classifier* if it can be represented by functions $f_i \in \mathcal{F}$ ($i = 1, \dots, r$) such that an input vector $\mathbf{x} \in \mathbb{A}$ is classified as positive ($\tau(\mathbf{x}) = \oplus$) iff $(f_1(\mathbf{x}) > t_1) \wedge \dots \wedge (f_r(\mathbf{x}) > t_r)$ for some constants t_i ($i = 1, \dots, r$). For example, if \mathcal{F} is the set of real-valued linear functions, then for \mathcal{F} -multi-threshold classifiers the set of positive examples \mathbf{x} is a polytope.

We are specifically interested in families of classifiers $\mathcal{T} \subseteq \mathcal{T}^{\mathcal{D}}$ which are closed under replacing arguments by constants (sometimes known as restriction or conditioning [15]) since this a necessary condition for the correctness of our polynomial-time algorithm. Fortunately, this is true for most families of functions of interest. For example, a linear/monotone/submodular threshold-classifier remains respectively linear/monotone/submodular if any of its arguments are replaced by constants. For $\tau \in \mathcal{T}^{\mathcal{D}}$ of arity n , $S \subseteq \{1, \dots, n\}$ and \mathbf{v} an assignment to the arguments indexed by S , let $\tau_{\mathbf{v}} : \prod_{i \notin S} D_i \rightarrow \mathcal{K}$ be the function obtained

from τ by fixing the arguments in S to \mathbf{v} , i.e. for all $\mathbf{x} \in \prod_{i \notin S} D_i$, $\tau_{\mathbf{v}}(\mathbf{x}) = \tau(\mathbf{v} \cup \mathbf{x})$. We say that \mathcal{T} is *closed under fixing arguments* if for all $\tau : \prod_{i=1}^n D_i \rightarrow \mathcal{K}$ such that $\tau \in \mathcal{T}$, for all $S \subseteq \{1, \dots, n\}$ and for all $\mathbf{v} \in \prod_{i \in S} D_i$, we have $\tau_{\mathbf{v}} \in \mathcal{T}$.

3 Tractability of finding one PI-explanation

To obtain a polynomial-time algorithm, we require that a particular decision problem be solvable in polynomial time. For a family $\mathcal{T} \subseteq \mathcal{T}^{\mathcal{D}}$ of boolean-valued functions, let $\text{TAUTOLOGY}(\mathcal{T})$ be the following decision problem: given a function $\tau \in \mathcal{T}$, is it true that $\tau \equiv \oplus$, i.e. for all $\mathbf{x} \in \mathbb{A}$, $\tau(\mathbf{x}) = \oplus$? To avoid exploring dead-end branches, our algorithm requires the answer to this question for functions obtained by fixing a subset of the arguments of a classifier, which is why we require that \mathcal{T} be closed under fixing arguments.

Firstly we consider the more general case in which the only assumption we make is that all functions in \mathcal{T} execute in polynomial time. In this case, $\text{TAUTOLOGY}(\mathcal{T}) \in \text{coNP}$ (since a counter-example can be verified in polynomial time). If, furthermore, \mathcal{T} is closed under fixing arguments, then using a greedy algorithm (as in Proposition 3.1 case (3) of [7]) we can deduce that n calls to an NP oracle are sufficient to find a PI-explanation. In the following, we investigate cases for which $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$ and hence for which finding a PI-explanation is also polynomial-time by a similar greedy algorithm.

We now state conditions which guarantee a polynomial-time algorithm to find one PI-explanation for large classes of classifiers. The algorithm initialises \mathcal{P} to \mathcal{A} and greedily deletes literals from \mathcal{P} as long as this preserves property (1) of being an explanation.

► **Proposition 4.** *If \mathcal{T} is closed under fixing arguments and $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$, then for any classifier $\tau \in \mathcal{T}$ and any positively-classified input \mathbf{a} , a PI-explanation of $\tau(\mathbf{a}) = \oplus$ can be found in polynomial time.*

Proof. An explanation is a set $\mathcal{P} \subseteq \{1, \dots, n\}$ such that equation (1) holds. The algorithm is a simple greedy algorithm that initialises \mathcal{P} to the trivial explanation $\{1, \dots, n\}$ (corresponding to the complete assignment \mathbf{a}) and for each $i \in \mathcal{P}$ tests whether i can be deleted to leave a valid explanation $\mathcal{P} \setminus \{i\}$:

$$\begin{aligned} \mathcal{P} &\leftarrow \{1, \dots, n\} \\ \text{for } i = 1, \dots, n : \\ &\quad \text{if } \mathcal{P} \setminus \{i\} \text{ is a valid explanation then } \mathcal{P} \leftarrow \mathcal{P} \setminus \{i\} \end{aligned}$$

Clearly, the final value $\tilde{\mathcal{P}}$ of \mathcal{P} is an explanation. Furthermore, it is minimal because if $\mathcal{P} \setminus \{i\}$ was not a valid explanation for some $\mathcal{P} \supseteq \tilde{\mathcal{P}}$, then neither is $\tilde{\mathcal{P}} \setminus \{i\}$.

Let \mathbf{v} be the partial assignment corresponding to the values a_j for $j \in \mathcal{P} \setminus \{i\}$. Testing whether $\mathcal{P} \setminus \{i\}$ is a valid explanation is equivalent to testing whether $\tau_{\mathbf{v}} \equiv \oplus$ and hence can be performed in polynomial time since \mathcal{T} is closed under fixing arguments and $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$. The algorithm needs to solve exactly n instances of $\text{TAUTOLOGY}(\mathcal{T})$. It follows that one PI-explanation can be found in polynomial time. ◀

Proposition 4 can be seen as a special case of the complexity of finding maximal solutions to problems for which the instance-solution relation is in P (Proposition 3.1 of [7]).

As we will now see, Proposition 4 applies to a large range of classifiers, such as linear, submodular or monotone threshold-classifiers as well as multi-threshold classifiers.

Consider threshold classifiers of the form $\tau(\mathbf{x}) = \oplus$ iff $f(\mathbf{x}) > t$, for some real-valued objective function $f \in \mathcal{F}$ and some constant t . Then

$$\tau \equiv \oplus \quad \Leftrightarrow \quad \min_{\mathbf{x} \in \mathbb{A}} f(\mathbf{x}) > t. \quad (2)$$

Thus, if \mathcal{T} is the set of \mathcal{F} -threshold classifiers, then $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$ if functions in \mathcal{F} can be minimised in polynomial time. Examples of classes of functions that can be minimised in polynomial time are the objective functions of extended linear classifiers (referred to as XLCs) [33], monotone functions over real/integer intervals [34] and submodular functions over finite ordered domains [31, 9].

Now consider the case of multi-threshold classifiers of the form $\tau(\mathbf{x}) = \oplus$ iff $\bigwedge_{i=1}^r f_i(\mathbf{x}) > t_i$, for some real-valued functions $f_i \in \mathcal{F}$ and some constants t_i ($i = 1, \dots, r$). Then

$$\tau \equiv \oplus \iff \bigwedge_{i=1}^r (\min_{\mathbf{x} \in \mathbb{A}} f_i(\mathbf{x}) > t_i). \quad (3)$$

Thus, if \mathcal{T} is the set of \mathcal{F} -multi-threshold classifiers, then again we have that $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$ if each function in \mathcal{F} can be minimised in polynomial time. For example, f_1 could be monotone, f_2 submodular and the other f_i linear.

We end this section by showing that a polytime tautology test is not only a sufficient but also a necessary condition for tractability of finding a PI-explanation. Let $\text{PIEXPL}^+(\mathcal{T})$ be the problem of finding a PI-explanation of a positive decision taken by a classifier in \mathcal{T} .

► **Theorem 5.** *If \mathcal{T} is closed under fixing arguments, then $\text{PIEXPL}^+(\mathcal{T}) \in \text{FP}$ iff $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$.*

Proof. The “if” part of the proof is Proposition 4. For the “only if” part, suppose that \mathcal{T} is closed under fixing arguments and $\text{PIEXPL}^+(\mathcal{T}) \in \text{FP}$. Let $\tau \in \mathcal{T}$. Let \mathbf{a} be an arbitrary choice of feature vector. Then τ is a tautology iff both $\tau(\mathbf{a}) = \oplus$ and the empty set is a PI-explanation of $\tau(\mathbf{a}) = \oplus$. Note that in the case that the empty set is a PI-explanation, it is necessarily the unique PI-explanation. Thus we can decide $\text{TAUTOLOGY}(\mathcal{T})$ in polynomial time. ◀

4 Explanations of negative decisions

In the previous section we exclusively studied the problem of finding an explanation of a *positive* decision $\tau(\mathbf{x}) = \oplus$. We show in this section that the complexity of this problem can change drastically if we require an explanation of a *negative* decision $\tau(\mathbf{x}) = \ominus$. For a family $\mathcal{T} \subseteq \mathcal{T}^{\mathcal{D}}$ of boolean functions, let $\text{UNSAT}(\mathcal{T})$ be the following decision problem: given a boolean function $\tau \in \mathcal{T}$, is it true that $\tau \equiv \ominus$, i.e. for all $\mathbf{x} \in \mathbb{A}$, $\tau(\mathbf{x}) = \ominus$? By an entirely similar proof based on a greedy algorithm, we can deduce the following proposition which mirrors Proposition 4.

► **Proposition 6.** *If \mathcal{T} is closed under fixing arguments and $\text{UNSAT}(\mathcal{T}) \in \text{P}$, then for any classifier $\tau \in \mathcal{T}$ and any negatively-classified input \mathbf{a} , a PI-explanation of $\tau(\mathbf{a}) = \ominus$ can be found in polynomial time.*

A simple case in which all features are boolean is \mathcal{T}_{DNF} , the family of DNF classifiers. Since deciding the (un)satisfiability of a DNF is trivial, we have $\text{UNSAT}(\mathcal{T}_{\text{DNF}}) \in \text{P}$ and so a PI-explanation of a negative decision can be found in polynomial time. On the other hand, by Theorem 5, and the co-NP-completeness of deciding whether a DNF is a tautology, a PI-explanation of a positive decision cannot be found in polynomial time (assuming $\text{P} \neq \text{NP}$).

Now consider threshold classifiers of the form $\tau(\mathbf{x}) = \oplus$ iff $f(\mathbf{x}) > t$, for some real-valued objective function $f \in \mathcal{F}$ and some constant t . Then

$$\tau \equiv \ominus \iff \max_{\mathbf{x} \in \mathbb{A}} f(\mathbf{x}) \leq t. \quad (4)$$

21:6 On the Tractability of Explaining Decisions of Classifiers

Thus, if \mathcal{T} is the set of \mathcal{F} -threshold classifiers, then $\text{UNSAT}(\mathcal{T}) \in \text{P}$ if functions in \mathcal{F} can be *maximised* in polynomial time. Examples of functions that can be maximised in polynomial time are linear, monotone, antitone (over real/integer intervals) or supermodular functions (over finite ordered domains). Note that submodular function maximisation cannot be achieved in polynomial time (assuming $\text{P} \neq \text{NP}$) [12].

Thus, for a given family of classifiers (such as submodular threshold classifiers), the complexity of finding an explanation of a positive decision may be polynomial-time whereas the complexity of finding an explanation of a negative decision may be intractable.

We end this section with a theorem that is the equivalent of Theorem 5 for negative decisions. Let $\text{PIEXPL}^-(\mathcal{T})$ be the problem of finding a PI-explanation of a negative decision taken by a classifier in \mathcal{T} .

► **Theorem 7.** *If \mathcal{T} is closed under fixing arguments, then $\text{PIEXPL}^-(\mathcal{T}) \in \text{FP}$ iff $\text{UNSAT}(\mathcal{T}) \in \text{P}$.*

Proof. The “if” part of the proof is Proposition 6. For the “only if” part, suppose that \mathcal{T} is closed under fixing arguments and $\text{PIEXPL}^-(\mathcal{T}) \in \text{FP}$. Let $\tau \in \mathcal{T}$. Let \mathbf{a} be an arbitrary choice of feature vector. Then τ is unsatisfiable iff both $\tau(\mathbf{a}) = \ominus$ and the empty set is a PI-explanation of $\tau(\mathbf{a}) = \ominus$. Thus we can decide $\text{UNSAT}(\mathcal{T})$ in polynomial time. ◀

5 Explanation of classifiers with constrained features

It may be that some constraints exist between features, so that not all vectors in \mathbb{A} are possible. For example, *gender = male* and *pregnant = yes* are incompatible, and clearly we must have *years_of_employment ≤ age*. This affects the definition of a PI-explanation. Suppose that there are constraints on the possible feature vectors \mathbf{x} given by a predicate $C(\mathbf{x})$. In the context of constraints C , a PI-explanation of a decision $\tau(\mathbf{a}) = c$ is now a subset-minimal set $\mathcal{P} \subseteq \mathcal{A}$ of feature literals such that

$$\forall (\mathbf{x} \in \mathbb{A}). \left(C(\mathbf{x}) \wedge \bigwedge_{j \in \mathcal{P}} (x_j = a_j) \right) \rightarrow \tau(\mathbf{x}) = c. \quad (5)$$

► **Example 8.** Consider a medicine that doctors are allowed to prescribe to everybody who has the flu except to pregnant women. A PI-explanation why Alice (who is pregnant) was not prescribed the medicine is that she is pregnant; there is no need to mention that she is a woman given the constraint that there are no pregnant men. There are two PI-explanations why Bob was prescribed the medicine: (1) that he is not pregnant and he had the flu, (2) that he is a man and he had the flu. Note that the rule for prescribing the medicine can be stated without mentioning gender: prescribe to people who have the flu but are not pregnant. The PI-explanations remain the same. In particular, the explanation (2) for Bob being prescribed the medicine mentions gender even though this feature is not mentioned in the rule. If we did not take into account the constraint that men cannot be pregnant, then the explanation (2) would not be valid.

Equating \ominus with 0 and \oplus with 1, we have the following equivalence which follows from equations (1), (5) and the logical equivalence $C \wedge A \rightarrow B \equiv A \rightarrow B \vee \neg C$

► **Proposition 9.** *A PI-explanation of a classifier τ under constraints C is precisely a PI-explanation of the unconstrained classifier $\tau \vee \neg C$.*

■ **Table 1** Examples of tractable families of constrained threshold-classifiers over finite domains.

decision	objective function f	constraints \mathcal{C}
positive	submodular	max and min-closed
positive	monotone	min-closed
positive	antitone	max-closed
negative	supermodular	max and min-closed
negative	monotone	max-closed
negative	antitone	min-closed

Consider a threshold classifier with objective function f under constraints \mathcal{C} . We can reduce to the unconstrained case by introducing the function g where

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathcal{C}(\mathbf{x}) \\ \infty & \text{if } \neg\mathcal{C}(\mathbf{x}). \end{cases} \quad (6)$$

Then a PI-explanation for $f(\mathbf{a}) > t$ under constraints \mathcal{C} is a PI-explanation of $f(\mathbf{a}) + g(\mathbf{a}) > t$ (in the unconstrained setting). We saw in Section 3 that finding a PI-explanation of a positive decision taken by a threshold classifier is polynomial-time if the objective function can be minimised in polynomial time. Thus, for example, if $f + g$ is submodular over finite domains, then a PI-explanation can be found in polynomial time. Assume in the following that f is finite-valued and g is defined as in equation (6). A necessary condition for $f + g$ to be submodular is that g be both min-closed and max-closed [10], where *min-closed* means $\mathcal{C}(\mathbf{x}) \wedge \mathcal{C}(\mathbf{y}) \Rightarrow \mathcal{C}(\min(\mathbf{x}, \mathbf{y}))$ and *max-closed* means $\mathcal{C}(\mathbf{x}) \wedge \mathcal{C}(\mathbf{y}) \Rightarrow \mathcal{C}(\max(\mathbf{x}, \mathbf{y}))$ [26]. Over finite domains, the class of monotone objective functions can be extended to a maximal tractable class of constrained minimisation problems by adding min-closed constraints and the class of antitone objective functions can be extended to a maximal tractable class by adding max-closed constraints [9].

As we have already seen, explanations of positive and negative decisions may have very different complexities. Indeed, a PI-explanation for $f(\mathbf{a}) \leq t$ under constraints \mathcal{C} is a PI-explanation of $f(\mathbf{a}) - g(\mathbf{a}) \leq t$ (in the unconstrained setting). The sign of g has changed so that the inequality is satisfied whenever g is infinite. As we saw in Section 4, a PI-explanation of a negative decision of a threshold classifier can be found in polynomial time if the objective function can be maximised in polynomial time. Thus, for example, if $f - g$ is a supermodular function (over finite domains), then a PI-explanation can be found in polynomial time. A necessary condition for $f - g$ to be supermodular is that g be both min-closed and max-closed [10]. For the class of monotone functions f , the maximisation of $f - g$ is tractable if the relations \mathcal{C} (corresponding to the functions g) are max-closed, and for the class of antitone functions f , the maximisation of $f - g$ is tractable if the relations \mathcal{C} are min-closed [9].

This allows us to identify the tractable families of constrained threshold-classifiers listed in Table 1.

6 Contrastive explanations

PI-explanations are also known as abductive explanations, since they are answers to the question “Why is $\tau(\mathbf{a}) = c$?” A contrastive explanation [35, 22, 21] is an answer to a different question: “Why is it not the case that $\tau(\mathbf{a}) \neq c$?” It gives a set of features which if changed in \mathbf{a} can lead to a change of class. Contrastive explanations tend to be smaller than abductive explanations and hence can be easier to interpret by a human user [35].

► **Definition 10.** Given that $\tau(\mathbf{a}) = c$, a contrastive explanation is a subset-minimal set $\mathcal{S} \subseteq \mathcal{A}$ such that

$$\exists(\mathbf{x} \in \mathbb{A}). \left(\left(\bigwedge_{j \notin \mathcal{S}} (x_j = a_j) \right) \wedge \tau(\mathbf{x}) \neq c \right). \quad (7)$$

If $\tau \equiv c$, then there is no contrastive explanation of $\tau(\mathbf{a}) = c$.

► **Example 11.** Consider the classifier studied in Example 2: a bank uses a function τ , given by $\tau(\mathbf{x}) = \oplus$ if and only if $(\max(\text{sal}_1, \text{sal}_2) \geq \text{sal}_{\min}) \wedge (\min(\text{age}_1, \text{age}_2) \leq \text{age}_{\max})$, to decide whether to grant a loan to a couple represented by a feature vector $\mathbf{x} = (\text{sal}_1, \text{sal}_2, \text{age}_1, \text{age}_2)$. If \mathbf{a} corresponds to a couple who both earn more than sal_{\min} and both are younger than age_{\max} , then the contrastive explanations of the decision $\tau(\mathbf{a}) = \oplus$ are $\{1, 2\}$ and $\{3, 4\}$. If \mathbf{b} corresponds to a couple who both earn more than sal_{\min} but both are older than age_{\max} , then the contrastive explanations of the decision $\tau(\mathbf{b}) = \ominus$ are $\{3\}$ and $\{4\}$.

Let $\text{INVALID}(\mathcal{T})$ be the following decision problem: given a boolean function $\tau \in \mathcal{T}$, does there exist $\mathbf{x} \in \mathbb{A}$ such that $\tau(\mathbf{x}) = \ominus$. Similarly, let $\text{SAT}(\mathcal{T})$ be the problem: given a boolean function $\tau \in \mathcal{T}$, does there exist $\mathbf{x} \in \mathbb{A}$ such that $\tau(\mathbf{x}) = \oplus$. The following proposition is the contrastive equivalent of Proposition 4 and Proposition 6.

► **Proposition 12.** Suppose that \mathcal{T} is closed under fixing arguments. If $\text{INVALID}(\mathcal{T}) \in \text{P}$, then for any classifier $\tau \in \mathcal{T}$ and any \mathbf{a} such that $\tau(\mathbf{a}) = \oplus$, a contrastive explanation of $\tau(\mathbf{a}) = \oplus$ can be found in polynomial time. If $\text{SAT}(\mathcal{T}) \in \text{P}$, then for any classifier $\tau \in \mathcal{T}$ and any \mathbf{a} such that $\tau(\mathbf{a}) = \ominus$, a contrastive explanation of $\tau(\mathbf{a}) = \ominus$ can be found in polynomial time.

Proof. We say that \mathcal{S} can lead to a class change if equation (7) holds. The algorithm is analogous to the algorithm for PI-explanations. It requires n tests of equation (7) to find a contrastive explanation:

```

 $\mathcal{S} \leftarrow \{1, \dots, n\}$ 
if  $\mathcal{S}$  cannot lead to a class change then report that no CXp exists ;
for  $i = 1, \dots, n$  :
    if  $\mathcal{S} \setminus \{i\}$  can lead to a class change then  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
    
```

Testing whether \mathcal{S} can lead to a class change from \oplus is a test of invalidity (after fixing features in $\mathcal{A} \setminus \mathcal{S}$), whereas testing whether \mathcal{S} can lead to a class change from \ominus is a test of satisfiability (after fixing features in $\mathcal{A} \setminus \mathcal{S}$). Thus, the above algorithm finds a contrastive explanation of $\tau(\mathbf{a}) = c$ in polynomial time if $\text{INVALID}(\mathcal{T}) \in \text{P}$ (in the case $c = \oplus$) or $\text{SAT}(\mathcal{T}) \in \text{P}$ (in the case $c = \ominus$). ◀

For threshold classifiers of the form $\tau(\mathbf{x}) = \oplus$ iff $f(\mathbf{x}) > t$, invalidity corresponds to $\min_{\mathbf{x} \in \mathbb{A}} f(\mathbf{x}) \leq t$ and satisfiability corresponds to $\max_{\mathbf{x} \in \mathbb{A}} f(\mathbf{x}) > t$. Thus, if \mathcal{T} is the set of \mathcal{F} -threshold classifiers, then $\text{INVALID}(\mathcal{T}) \in \text{P}$ if functions in \mathcal{F} can be minimised in polynomial time and $\text{SAT}(\mathcal{T}) \in \text{P}$ if functions in \mathcal{F} can be maximised in polynomial time.

Let $\text{CEXPL}^+(\mathcal{T})$ (respectively, $\text{CEXPL}^-(\mathcal{T})$) be the problem of finding a contrastive explanation of a positive (negative) decision taken by a classifier in \mathcal{T} or determining that no contrastive explanation exists. The following theorem follows from Proposition 12 and the fact that deciding the existence of a contrastive explanation of $\tau(\mathbf{a}) = c$ is equivalent to deciding $\neg(\tau \equiv c)$.

► **Theorem 13.** If \mathcal{T} is closed under fixing arguments, then $\text{CEXPL}^+(\mathcal{T}) \in \text{FP}$ iff $\text{INVALID}(\mathcal{T}) \in \text{P}$, and $\text{CEXPL}^-(\mathcal{T}) \in \text{FP}$ iff $\text{SAT}(\mathcal{T}) \in \text{P}$.

In the context of constraints C , a contrastive explanation of a decision $\tau(\mathbf{a}) = c$ is now a subset-minimal set $\mathcal{S} \subseteq \mathcal{A}$ of feature literals such that

$$\exists(\mathbf{x} \in \mathbb{A}). \left(\left(\bigwedge_{j \notin \mathcal{S}} (x_j = a_j) \right) \wedge \tau(\mathbf{x}) \neq c \wedge C(\mathbf{x}) \right). \quad (8)$$

Equating \ominus with 0 and \oplus with 1, and using the logical equivalence $\neg B \wedge C \equiv \neg(B \vee \neg C)$, we have the following proposition.

► **Proposition 14.** *A contrastive explanation of a classifier τ under constraints C is precisely a contrastive explanation of the unconstrained classifier $\tau \vee \neg C$.*

In the case of constrained threshold classifiers, with objective function f and threshold t , let g be as defined by equation (6). Then testing invalidity under constraints C is equivalent to determining whether $\min_{\mathbf{x} \in \mathbb{A}} (f(\mathbf{x}) + g(\mathbf{x})) \leq t$ and testing satisfiability is equivalent to determining whether $\max_{\mathbf{x} \in \mathbb{A}} (f(\mathbf{x}) - g(\mathbf{x})) > t$. It follows that the tractable cases for finding contrastive explanations or PI-explanations are identical. Example are shown in Table 1, where, in both cases, the decision corresponds to the original decision (i.e. the value of $\tau(\mathbf{a})$).

In fact, from Theorem 5, Theorem 7, Theorem 13, Proposition 9 and Proposition 14, we can deduce the following theorem which says that tractable classes of finding abductive or contrastive explanations coincide. It follows from the fact that $\text{INVALID}(\mathcal{T}) \in \text{P}$ iff $\text{TAUTOLOGY}(\mathcal{T}) \in \text{P}$ and that $\text{SAT}(\mathcal{T}) \in \text{P}$ iff $\text{UNSAT}(\mathcal{T}) \in \text{P}$ (since a problem is in P iff its complement is in P).

► **Theorem 15.** *In the unconstrained or constrained setting, if \mathcal{T} is closed under fixing arguments, $\text{PIEXPL}^+(\mathcal{T}) \in \text{FP}$ iff $\text{CEXPL}^+(\mathcal{T}) \in \text{FP}$, and $\text{PIEXPL}^-(\mathcal{T}) \in \text{FP}$ iff $\text{CEXPL}^-(\mathcal{T}) \in \text{FP}$.*

7 A language dichotomy for threshold classifiers

We consider threshold classifiers over finite (i.e. categorical) domains whose objective function can be decomposed into functions of bounded arity:

$$f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}[\sigma_i]) \quad (9)$$

where each σ_i (the scope on which the function f_i is applied) is a list of indices from $\{1, \dots, n\}$ and $\mathbf{x}[\sigma_i]$ is the projection of the vector \mathbf{x} on these indices. Given a set (language) \mathcal{L} of functions, we denote by $\mathcal{T}_{\mathcal{L}}$ the set of threshold classifiers whose objective function f is the sum of functions $f_i \in \mathcal{L}$. Recall that $\text{PIEXPL}^+(\mathcal{T}_{\mathcal{L}})$ is the problem of finding a PI-explanation of a positive decision taken by a classifier in $\mathcal{T}_{\mathcal{L}}$.

Cost Function Networks (CFNs) (also known as Valued Constraint Satisfaction Problems) are defined by sets of functions f_i (and their associated scopes) over finite domains whose sum f (given by equation (9)) is an objective function to be minimized [11]. CFNs are a generic framework covering many well-studied optimisation problems. For example, Bayesian networks can be transformed into CFNs after taking logarithms of probabilities [11]. Let $\text{CFN}(\mathcal{L})$ denote the problem of determining, given an objective function f of the form given in equation (9) where each $f_i \in \mathcal{L}$, together with a real constant t , whether

$$\min f(\mathbf{x}) \leq t.$$

A technical point is that, due to the necessarily bounded precision of the values of functions, this is equivalent to the problem of determining, given f and $t \in \mathbb{R}$, whether $\min f(\mathbf{x})$ is strictly less than t .

21:10 On the Tractability of Explaining Decisions of Classifiers

The complexity of $\text{CFN}(\mathcal{L})$ has been extensively studied for finite languages (i.e. languages \mathcal{L} such that $|\mathcal{L}|$ is finite). It is now known that there is a dichotomy: depending on the language \mathcal{L} , $\text{CFN}(\mathcal{L})$ is either in P or is NP-complete. This result was known for languages of finite-valued cost-functions [41] and the dichotomy for the more general case, in which costs can be infinite, follows from the recently-discovered language dichotomy for constraint satisfaction problems [4, 44, 29, 30]. The following proposition will lead us to a similar dichotomy for explaining decisions.

► **Proposition 16.** *Let \mathcal{L} be a set of non-negative functions closed under fixing arguments. Then $\text{PIEXPL}^+(\mathcal{T}_{\mathcal{L}})$ is in FP if and only if $\text{CFN}(\mathcal{L})$ is in P.*

Proof. If \mathcal{L} is closed under fixing arguments, then so is $\mathcal{T}_{\mathcal{L}}$. The “if” part of the proof follows directly from Proposition 4 and the subsequent discussion in Section 3, so we concentrate on the “only if” part.

By Theorem 5 we know that if $\text{PIEXPL}^+(\mathcal{T}_{\mathcal{L}}) \in \text{FP}$ then $\text{TAUTOLOGY}(\mathcal{T}_{\mathcal{L}}) \in \text{P}$. $\text{TAUTOLOGY}(\mathcal{T}_{\mathcal{L}})$ is the problem of determining, for a function f expressible as the sum of functions $f_i \in \mathcal{L}$ (as in equation (9)) and a constant t , whether $f(x) > t$ for all $x \in \mathbb{A}$. This is the complement of $\text{CFN}(\mathcal{L})$ which is the problem of determining whether $\min_{x \in \mathbb{A}} f(x) \leq t$. Hence, if $\text{TAUTOLOGY}(\mathcal{T}_{\mathcal{L}}) \in \text{P}$ then $\text{CFN}(\mathcal{L}) \in \text{P}$, which completes the proof. ◀

We now consider constrained classifiers. Let Γ be a language of constraint relations. For each constraint relation in Γ we can construct a corresponding $\{0, \infty\}$ -valued function g , as given by equation (6). Let \mathcal{C}_{Γ} denote the set of all such $\{0, \infty\}$ -valued functions for relations in Γ . Then $\mathcal{L} \cup \mathcal{C}_{\Gamma}$ can be viewed as a language of cost functions. Let $\text{CONPIEXPL}^+(\mathcal{T}_{\mathcal{L}}, \Gamma)$ (respectively, $\text{CONPIEXPL}^-(\mathcal{T}_{\mathcal{L}}, \Gamma)$) denote the problem of finding one PI-explanation of a positive (negative) decision taken by a classifier in $\mathcal{T}_{\mathcal{L}}$ under a finite set of constraints from Γ .

► **Proposition 17.** *Let \mathcal{L} be a set of non-negative functions closed under fixing arguments and Γ a finite set of constraint relations. Then $\text{CONPIEXPL}^+(\mathcal{T}_{\mathcal{L}}, \Gamma)$ is in FP if and only if $\text{CFN}(\mathcal{L} \cup \mathcal{C}_{\Gamma})$ is in P.*

Proof. We know from the discussion in Section 5 that $\text{CONPIEXPL}^+(\mathcal{T}_{\mathcal{L}}, \Gamma)$ is equivalent to $\text{PIEXPL}^+(\mathcal{T}_{\mathcal{L} \cup \mathcal{C}_{\Gamma}})$. Thus the result follows immediately from Proposition 16. ◀

We now consider finding explanations for negative decisions. Although, as we will show, there is again a dichotomy, it is not the same since in this case we are studying a (constrained) maximisation problem rather than a (constrained) minimisation problem. Given a finite language \mathcal{L} of real-valued functions, all bounded above by $B \in \mathbb{R}$, let \mathcal{L}_{inv} denote the set $\{B - f : f \in \mathcal{L}\}$. Clearly, maximising a sum of functions from \mathcal{L} is equivalent to minimising a sum of functions from \mathcal{L}_{inv} .

► **Proposition 18.** *Let \mathcal{L} be a set of non-negative finite-valued functions closed under fixing arguments. Then $\text{PIEXPL}^-(\mathcal{T}_{\mathcal{L}})$ is in FP if and only if $\text{CFN}(\mathcal{L}_{\text{inv}})$ is in P.*

Proof. The “if” part follows from Proposition 6 and the subsequent discussion in Section 4. For the “only if” part, we know from Theorem 7 that if $\text{PIEXPL}^-(\mathcal{T}_{\mathcal{L}})$ is in FP then $\text{UNSAT}(\mathcal{T}_{\mathcal{L}})$ is in P. $\text{UNSAT}(\mathcal{T}_{\mathcal{L}})$ is the problem of determining, for a function f expressible as the sum of m functions $f_i \in \mathcal{L}$ and a constant t , whether $f(x) \leq t$ for all $x \in \mathbb{A}$. This is equivalent to determining whether $mB - f(x) \geq mB - t$ for all $x \in \mathbb{A}$. This is the complement of the problem of determining whether $\min(mB - f) < t'$ (for $t' = mB - t$). This is precisely $\text{CFN}(\mathcal{L}_{\text{inv}})$. Hence, if $\text{UNSAT}(\mathcal{T}_{\mathcal{L}}) \in \text{P}$, then $\text{CFN}(\mathcal{L}_{\text{inv}}) \in \text{P}$, which completes the proof. ◀

We now generalise this result to constrained classifiers.

► **Proposition 19.** *Let \mathcal{L} be a set of non-negative functions closed under fixing arguments and Γ a finite set of constraint relations. Then $\text{CONPIEXPL}^-(\mathcal{T}_{\mathcal{L}}, \Gamma)$ is in FP if and only if $\text{CFN}(\mathcal{L}_{\text{inv}} \cup \mathcal{C}_{\Gamma})$ is in P.*

Proof. It is easy to see that $\text{CONPIEXPL}^-(\mathcal{T}_{\mathcal{L}}, \Gamma)$ is equivalent to $\text{CONPIEXPL}^+(\mathcal{T}_{\mathcal{L}_{\text{inv}}}, \Gamma)$. Thus the result follows immediately from Proposition 17. ◀

Given the known P/NP-complete dichotomy for $\text{CFN}(\mathcal{L})$ for finite languages \mathcal{L} , discussed above, we can immediately deduce the following theorem.

► **Theorem 20.** *Let \mathcal{L} be a finite language of non-negative functions closed under fixing arguments and Γ a finite set of constraint relations. Then each of $\text{PIEXPL}^+(\mathcal{T}_{\mathcal{L}})$, $\text{CONPIEXPL}^+(\mathcal{T}_{\mathcal{L}}, \Gamma)$, $\text{PIEXPL}^-(\mathcal{T}_{\mathcal{L}})$, $\text{CONPIEXPL}^-(\mathcal{T}_{\mathcal{L}}, \Gamma)$ is either in FP or is NP-hard.*

Indeed, by Theorem 15, we have an identical dichotomy result for contrastive explanations.

8 Diversity of explanations

We have concentrated up until now on the problem of finding a single explanation. This is because the problem of finding all explanations has the obvious disadvantage that the number of explanations may be exponential. For example, in a first-past-the-post election in which a A wins with $m \geq k$ out of the $n = 2k - 1$ votes cast, and each vote is considered as a feature, there are C_m^k PI-explanations for this victory; for a candidate B who lost with only $p \leq k$ votes, there are C_{n-p}^{k-p} contrastive explanations for why they did not win.

Rather than providing a single explanation to the user or listing all explanations, we can envisage providing a relatively small number of *diverse* explanations. A similar strategy of finding a number of diverse good-quality solutions to a Weighted Constraint Satisfaction Problem has been used successfully in computational protein design [37], among other examples [18, 19, 25].

An obvious measure of diversity of a set of explanations $\{S_1, \dots, S_k\}$ is the minimum Hamming distance $|S_i \Delta S_j|$ between pairs of distinct explanations S_i, S_j , where Δ is the symmetric difference operator between two sets. This leads to the following computational problem.

k -DIV-PIEXPL⁺: Given a binary classifier $\tau : \mathbb{A} \rightarrow \{\ominus, \oplus\}$, a positively-classified input \mathbf{a} and an integer m , find k PI-explanations S_1, \dots, S_k of $\tau(\mathbf{a}) = \oplus$ such that for all i, j such that $1 \leq i < j \leq k$, $|S_i \Delta S_j| \geq m$.

The definitions for negatively-classified inputs \mathbf{a} (k -DIV-PIEXPL⁻) and/or for contrastive explanations (k -DIV-CEXPL⁺, k -DIV-CEXPL⁻) are entirely similar. Since Hamming distance is a submodular function, one might hope that there would be interesting tractable classes. Unfortunately, since we are, in a sense, maximising this distance rather than minimising it, these four problems turn out to be NP-hard even in the simplest non-trivial case.

► **Proposition 21.** *Even in the case of $k = 2$ and for a linear classifier τ over domains of size 2, the following four problems are NP-hard: (a) k -DIV-PIEXPL⁺, (b) k -DIV-PIEXPL⁻, (c) k -DIV-CEXPL⁺, (d) k -DIV-CEXPL⁻.*

Proof.

(a) Without loss of generality, we suppose that the domains D_i ($i = 1, \dots, n$) are all $\{0, 1\}$ and $\tau(\mathbf{x}) = \oplus$ iff $\sum_{i=1}^n \alpha_i x_i > t$. We prove NP-hardness for the particular case in which $\mathbf{a} = (1, \dots, 1)$ and the values $t, \alpha_1, \dots, \alpha_n$ are strictly positive integers which satisfy the following inequalities:

21:12 On the Tractability of Explaining Decisions of Classifiers

$$\alpha_1 \leq \dots \leq \alpha_m < \alpha_{m+1} \leq \dots \leq \alpha_n \quad (10)$$

$$\sum_{i=1}^m \alpha_i + 2 \sum_{i=m+1}^n \alpha_i = 2(t+1). \quad (11)$$

To solve 2-DIV-PIEXPL⁺ we require sets $S_1, S_2 \subseteq \{1, \dots, n\}$ satisfying (1) $|S_1 \Delta S_2| \geq m$ and (2) S_1, S_2 are minimal (for inclusion) sets such that the minimum value of $\sum_{i=1}^n \alpha_i x_i$ is at least $t+1$ for inputs \mathbf{x} with $x_i = \alpha_i = 1$ for all $i \in S_j$ ($j = 1, 2$). Since the values α_i are positive, the minimum is attained when $x_i = 0$ for all $i \notin S_j$, and so this is equivalent to

$$\sum_{i \in S_j} \alpha_i \geq t+1 \quad (j = 1, 2). \quad (12)$$

Summing these two inequalities (for $j = 1, 2$) gives

$$\sum_{i \in S_1} \alpha_i + \sum_{i \in S_2} \alpha_i \geq 2(t+1). \quad (13)$$

Since, by (10), we have $\alpha_r < \alpha_s$ for $r \leq m < s$, and $|S_1 \Delta S_2| \geq m$, we know that the left hand side of the sum in equation (13) is at most equal to the left hand side of equation (11), which is equal to $2(t+1)$. It follows that we actually have equality in inequality (13) and $S_1 \Delta S_2 = \{1, \dots, m\}$ and $S_1 \cap S_2 = \{m+1, \dots, n\}$. Equality in (13) implies that we must also have equality in the inequalities (12) for $j = 1, 2$. Equality implies minimality for subset inclusion since all weights α_i are strictly positive. Denoting $t+1 - \sum_{i=m+1}^n \alpha_i$ by T and $S_j \cap \{1, \dots, m\}$ by P_j (for $j = 1, 2$), we can deduce that we require a partition P_1, P_2 of $\{1, \dots, m\}$ such that

$$\sum_{i \in P_1} \alpha_i = T = \sum_{i \in P_2} \alpha_i.$$

This is precisely the partition problem which is well known to be NP-complete [28]. It follows that k -DIV-PIEXPL⁺ is NP-hard.

- (b) We consider the same linear classifier τ as in case (a), except that equation (11) is replaced by $\sum_{i=1}^m \alpha_i = 2t$, and this time we consider the vector $\mathbf{a} = (0, \dots, 0)$ which is classified negatively by τ . To solve k -DIV-PIEXPL⁻, we require two sets S_1, S_2 such that (1) $|S_1 \Delta S_2| \geq m$ and (2) S_1, S_2 are minimal (for inclusion) sets such that $\sum_{i \notin S_j} \alpha_i \leq t$ ($j = 1, 2$). Given equation (10), this can only be attained when $S_1 \Delta S_2 = \{1, \dots, m\}$ and $S_1 \cap S_2 = \{m+1, \dots, n\}$, so that $\sum_{i \notin S_1} \alpha_i = \sum_{i \notin S_2} \alpha_i = t$. Thus, we need to find two sets $P_j = \{1, \dots, n\} \setminus S_j$ ($j = 1, 2$) which partition $\{1, \dots, m\}$ and such that

$$\sum_{i \in P_1} \alpha_i = t = \sum_{i \in P_2} \alpha_i$$

Thus, again we have a polynomial reduction from the partition problem. Hence k -DIV-PIEXPL⁻ is NP-hard.

- (c) Consider the same linear classifier τ as in case (b), but this time $\mathbf{a} = (1, \dots, 1)$. To solve k -DIV-CEXPL⁺, we require two sets $S_1, S_2 \subseteq \{1, \dots, n\}$ such that $\sum_{i \notin S_j} \alpha_i \leq t$ ($j = 1, 2$) and $|S_1 \Delta S_2| \geq m$. Since this is exactly the same problem encountered in case (b), we can again deduce NP-hardness.

- (d) Consider the same linear classifier τ as in case (a), but with $\mathbf{a} = (0, \dots, 0)$. To solve $k\text{-DIV-CEXPL}^-$, we require $S_1, S_2 \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S_j} \alpha_i \geq t + 1$ ($j = 1, 2$) and $|S_1 \Delta S_2| \geq m$. Since this is exactly the problem encountered in case (a), we can again deduce NP-hardness. \blacktriangleleft

It is well known that the partition problem is one of the easiest NP-hard problems to solve in practice [38]. Thus, Proposition 21 precludes (assuming $P \neq NP$) a worst-case polynomial-time algorithm for finding a diverse set of explanations, but leaves the door open to the existence of practically-efficient algorithms.

9 Absolute explanations

Given a classifier we may want to have an absolute (global) explanation for a given class c , rather than an explanation specific to a particular decision. This answers questions of the type “Why can a customer be granted (or refused) a loan”. An absolute explanation is a minimal but arbitrary partial assignment to the features that guarantees that the output of the classifier τ will be the class c [24]. It does not depend on a concrete input instance but rather the entire feature space.

A *literal* is an assignment of a value to a feature which we can write in the form $(x_i = u)$ or simply as the pair $\langle i, u \rangle$ where $i \in \mathcal{A}$ and u belongs to D_i the domain of feature i . A set of literals \mathcal{U} is *well-defined* if each feature i occurs at most once in \mathcal{U} . For simplicity of presentation, we implicitly assume from now on that all subsets of literals are well-defined. This means that each subset of literals \mathcal{U} corresponds to a partial assignment \mathbf{a} to some subset of features $\mathcal{P} \subseteq \mathcal{A}$. This allows us to equate \mathcal{U} with the pair $\langle \mathcal{P}, \mathbf{a} \rangle$.

► **Definition 22.** *Given a classifier τ , an absolute explanation (XP) for a class c is a subset-minimal set of literals $\mathcal{U} = \langle \mathcal{P}, \mathbf{a} \rangle$ such that*

$$\forall (\mathbf{x} \in \mathbb{A}). \bigwedge_{j \in \mathcal{P}} (x_j = a_j) \rightarrow \tau(\mathbf{x}) = c \quad (14)$$

is true.

Equation (14) is identical to equation (1) in the definition of a PI-explanation, the difference being that an XP is a set of literals rather than a set of features. A subtle difference between PI-explanations and XP’s is that whereas a PI-explanation always exists, since we are given an instance \mathbf{a} such that $\tau(\mathbf{a}) = c$, an XP may not exist (which corresponds to the case when τ never takes the value c).

Associating a set of literals with the term corresponding to their conjunction, we can observe that a model τ is logically equivalent to the disjunction of the absolute explanations for the class \oplus . This observation shows that, in the case of finite domains, a black-box model can in theory be reconstructed from its absolute explanations.

Another global notion, dual to the notion of absolute explanation, is that of a counterexample [24]. This is an answer to questions such as “Why can a customer not be granted a loan”.

► **Definition 23.** *Given a classifier τ , a counterexample (CEX) for a class c is a subset-minimal set of literals $\mathcal{U} = \langle \mathcal{P}, \mathbf{a} \rangle$ such that*

$$\forall (\mathbf{x} \in \mathbb{A}). \bigwedge_{j \in \mathcal{P}} (x_j = a_j) \rightarrow \tau(\mathbf{x}) \neq c \quad (15)$$

is true.

21:14 On the Tractability of Explaining Decisions of Classifiers

Clearly, in the case of binary classifiers with $\mathcal{K} = \{\ominus, \oplus\}$, a counterexample for class \oplus (\ominus) is an absolute explanation of class \ominus (\oplus). Analogously to the fact, observed above, that a model τ is logically equivalent to the disjunction of the absolute explanations for the class \oplus , it is also logically equivalent to the conjunction of the negations of the counterexamples of the class \oplus .

► **Example 24.** We return to the function τ used by a bank to decide whether to grant a loan to a couple represented by a feature vector $\mathbf{x} = (sal_1, sal_2, age_1, age_2)$, as in Example 2: $\tau(\mathbf{x}) = \oplus$ if and only if $(\max(sal_1, sal_2) \geq sal_{\min}) \wedge (\min(age_1, age_2) \leq age_{\max})$. If $s_0 \geq sal_{\min} > s_1, s_2$ and $a_0 \leq age_{\max}$, then $\{\langle 1, s_0 \rangle, \langle 3, a_0 \rangle\}$ is an XP of a positive decision (and a CEx of a negative decision) whereas $\{\langle 1, s_1 \rangle, \langle 2, s_2 \rangle\}$ is an XP of a negative decision (and a CEx of a positive decision).

Despite the similarity between the definitions of PI-explanations and absolute explanations, the complexity of finding one XP is not the same as the complexity of finding one PI-explanation. This is due to the fact that we are not given a specific instance, but rather a class c , and we actually have to find an instance which belongs to class c . Let $XP^+(\mathcal{T})$ (respectively, $XP^-(\mathcal{T})$) be the problem of finding an absolute explanation of a positive (negative) decision taken by a classifier in \mathcal{T} (or returning “none” if none exists).

► **Theorem 25.** *If \mathcal{T} is closed under fixing arguments, and domains of all features are finite, then $XP^+(\mathcal{T}) \in FP$ iff $SAT(\mathcal{T}) \in P$ and $TAUTOLOGY(\mathcal{T}) \in P$.*

Proof. For the “if” part, it is sufficient to give a polynomial-time algorithm. Consider $\tau \in \mathcal{T}$. A call to $SAT(\mathcal{T})$ tells us whether or not an XP exists. In the case that an XP exists, we can find an instance \mathbf{a} such that $\tau(\mathbf{a}) = \oplus$ by the following incremental algorithm.

```

Initialise  $\mathbf{a}$  to the empty assignment ;
for  $i = 1, \dots, n$  :
    for each value  $d \in D_i$ 
        extend  $\mathbf{a}$  by assigning  $a_i = d$ ;
        if  $\tau_{\mathbf{a}}$  is satisfiable then exit the inner for loop;

```

The partial assignment \mathbf{a} is initialised to the empty assignment and successively, for each feature i , at most $|D_i|$ calls to $SAT(\mathcal{T})$ are sufficient to find a value for a_i which extends the present partial assignment so that $\tau_{\mathbf{a}}$ remains satisfiable. The final value of \mathbf{a} is a complete assignment such that $\tau(\mathbf{a}) = \oplus$. Since $TAUTOLOGY(\mathcal{T}) \in P$, by Proposition 4 we can find a PI-explanation \mathcal{P} of $\tau(\mathbf{a}) = \oplus$ in polynomial time. Then $\langle \mathcal{P}, \mathbf{a}[\mathcal{P}] \rangle$ is necessarily an XP, where $\mathbf{a}[\mathcal{P}]$ is the partial assignment of \mathbf{a} on features \mathcal{P} .

For the “only if” part, an XP exists iff $\tau \not\equiv \ominus$ and is non-empty iff $\tau \not\equiv \oplus$. Hence, a polynomial-time algorithm for $XP^+(\mathcal{T})$ necessarily decides both $SAT(\mathcal{T})$ and $TAUTOLOGY(\mathcal{T})$ in polynomial time. ◀

► **Corollary 26.** *If \mathcal{T} is closed under fixing arguments, and domains of all features are finite, then $XP^-(\mathcal{T}) \in FP$ iff $SAT(\mathcal{T}) \in P$ and $TAUTOLOGY(\mathcal{T}) \in P$.*

Proof. First, observe that an absolute explanation (XP) of a negative decision taken by a classifier τ is an XP of a positive decision taken by the classifier $\bar{\tau}$. To complete the proof, it suffices to notice that $\bar{\tau}$ is satisfiable iff τ is not a tautology (and $\bar{\tau}$ is a tautology iff τ is not satisfiable). So, by Theorem 25, $XP^-(\mathcal{T}) \in FP$ iff $SAT(\mathcal{T}) \in P$ and $TAUTOLOGY(\mathcal{T}) \in P$. ◀

For a family \mathcal{T} of threshold classifiers, Theorem 25 implies that $XP^+(\mathcal{T}) \in FP$ iff the corresponding family of objective functions can be both minimized and maximized in polynomial time. Examples are monotone functions and modular functions. Modular functions, which by definition are both submodular and supermodular, are separable (i.e. expressible as the sum of unary functions on the features) [42].

In the case of constrained classifiers we have the following proposition which follows directly from Proposition 9.

► **Proposition 27.** *An absolute explanation (XP) of a classifier τ under constraints C is precisely an XP of the unconstrained classifier $\tau \vee \neg C$.*

With f the objective function of the threshold classifier τ and g the function, given by Equation 6, associated with the constraints C , $\text{TAUTOLOGY}(\mathcal{T})$ corresponds to minimizing $f + g$ and $\text{SAT}(\mathcal{T})$ corresponds to maximizing $f - g$. By Theorem 25 together with the discussion above and in Section 5, $XP^+(\mathcal{T})$ is tractable for objective functions f which are either modular, monotone or antitone and constraint relations C which are both min and max-closed.

10 Discussion and Conclusion

We have investigated the complexity of finding one subset-minimal abductive or contrastive explanation for different families of classifiers.

There remain many interesting open questions:

- Since, as yet, there is no known characterisation of the complexity of cost-function languages over infinite domains, the complexity of classifiers with real-valued features is still an open problem.
- We have investigated the problem of finding a subset-minimal explanation. The problem of finding a cardinality-minimum explanation is naturally harder [39, 3] even though it has been observed that there is often not a significant difference between the size of subset-minimal and cardinality-minimum explanations [23]. It is known that the problem of finding a cardinality-minimum explanation is NP-hard for decision trees [3] and is tractable for linear classifiers [33]. It is an open theoretical question whether there are any other interesting tractable cases.
- Instead of searching for one explanation, we may want to find many explanations. Unfortunately, the fact that a greedy algorithm can find one explanation in polynomial time provides no guarantee that explanations can be enumerated in polynomial delay. Again, for linear classifiers, there is a polynomial-delay algorithm for enumerating PI-explanations [33], and it is an open question whether this is true for other families of classifiers. It is known to be false for monotone classifiers (assuming $P \neq NP$) [34].

References

- 1 Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the computational intelligibility of boolean classifiers. *CoRR*, abs/2104.06172, 2021. [arXiv:2104.06172](https://arxiv.org/abs/2104.06172).
- 2 Gilles Audemard, Frédéric Koriche, and Pierre Marquis. On tractable XAI queries based on compiled representations. In *KR*, pages 838–849, 2020. [doi:10.24963/kr.2020/86](https://doi.org/10.24963/kr.2020/86).

- 3 Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. Model interpretability through the lens of computational complexity. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS 2020*, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/b1adda14824f50ef24ff1c05bb66faf3-Abstract.html>.
- 4 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *FOCS*, pages 319–330, 2017. doi:10.1109/FOCS.2017.37.
- 5 Rainer E. Burkard, Bettina Klinz, and Rüdiger Rudolf. Perspectives of Monge properties in optimization. *Discret. Appl. Math.*, 70(2):95–161, 1996.
- 6 Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic submodular function minimization. In *STOC*, pages 1220–1231, 2017.
- 7 Zhi-Zhong Chen and Seinosuke Toda. The complexity of selecting maximal solutions. *Inf. Comput.*, 119(2):231–239, 1995. doi:10.1006/inco.1995.1087.
- 8 David A. Cohen, Martin C. Cooper, Peter Jeavons, and Andrei A. Krokhin. A maximal tractable class of soft constraints. *J. Artif. Intell. Res.*, 22:1–22, 2004.
- 9 David A. Cohen, Martin C. Cooper, Peter Jeavons, and Andrei A. Krokhin. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006.
- 10 Martin C. Cooper, Simon de Givry, Martí Sánchez-Fibla, Thomas Schiex, Matthias Zytnicki, and Tomás Werner. Soft arc consistency revisited. *Artif. Intell.*, 174(7-8):449–478, 2010.
- 11 Martin C. Cooper, Simon de Givry, and Thomas Schiex. Graphical models: Queries, complexity, algorithms (tutorial). In *STACS*, pages 4:1–4:22, 2020.
- 12 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of Boolean constraint satisfaction problems*, volume 7 of *SIAM monographs on discrete mathematics and applications*. SIAM, 2001.
- 13 Adnan Darwiche. Three modern roles for logic in AI. In *PODS*, pages 229–243, 2020. doi:10.1145/3375395.3389131.
- 14 Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *ECAI*, pages 712–720, 2020. doi:10.3233/FAIA200158.
- 15 Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264, 2002. doi:10.1613/jair.989.
- 16 Satoru Fujishige. *Submodular Functions and Optimisation*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, 2nd edition, 2005.
- 17 Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019. doi:10.1145/3236009.
- 18 Emmanuel Hebrard, Brahim Hnich, Barry O'Sullivan, and Toby Walsh. Finding diverse and similar solutions in constraint programming. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence*, pages 372–377. AAAI Press / The MIT Press, 2005. URL: <http://www.aaai.org/Library/AAAI/2005/aaai05-059.php>.
- 19 John Horan and Barry O'Sullivan. Towards diverse relaxations of over-constrained models. In *ICTAI 2009, 21st IEEE International Conference on Tools with Artificial Intelligence*, pages 198–205. IEEE Computer Society, 2009. doi:10.1109/ICTAI.2009.89.
- 20 Alexey Ignatiev. Towards trustable explainable AI. In *IJCAI*, pages 5154–5158, 2020. doi:10.24963/ijcai.2020/726.
- 21 Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva. From contrastive to abductive explanations and back again. In Matteo Baldoni and Stefania Bandini, editors, *AIXIA 2020*, volume 12414 of *Lecture Notes in Computer Science*, pages 335–355. Springer, 2020. doi:10.1007/978-3-030-77091-4_21.
- 22 Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva. On relating ‘why?’ and ‘why not?’ explanations. *CoRR*, abs/2012.11067, 2020. arXiv:2012.11067.

- 23 Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-based explanations for machine learning models. In *AAAI*, pages 1511–1519, 2019. doi:10.1609/aaai.v33i01.33011511.
- 24 Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. On relating explanations and adversarial examples. In *NeurIPS*, pages 15857–15867, 2019. URL: <http://papers.nips.cc/paper/9717-on-relating-explanations-and-adversarial-examples>.
- 25 Linnea Ingmar, Maria Garcia de la Banda, Peter J. Stuckey, and Guido Tack. Modelling diversity of solutions. In *AAAI 2020*, pages 1528–1535. AAAI Press, 2020. URL: <https://aaai.org/ojs/index.php/AAAI/article/view/5512>.
- 26 Peter Jeavons and Martin C. Cooper. Tractable constraints on ordered domains. *Artif. Intell.*, 79(2):327–339, 1995. doi:10.1016/0004-3702(95)00107-7.
- 27 Matthew Joseph, Michael J. Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS 2016*, pages 325–333, 2016. URL: <https://proceedings.neurips.cc/paper/2016/hash/eb163727917cbb1ee208541a643e74-Abstract.html>.
- 28 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.
- 29 Vladimir Kolmogorov, Andrei A. Krokhnin, and Michal Rolínek. The complexity of general-valued CSPs. *SIAM J. Comput.*, 46(3):1087–1110, 2017. doi:10.1137/16M1091836.
- 30 Andrei A. Krokhnin and Stanislav Zivný. The complexity of valued CSPs. In Andrei A. Krokhnin and Stanislav Zivný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 233–266. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/DFU.Vol7.15301.9.
- 31 Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *FOCS*, pages 1049–1065, 2015.
- 32 Xingchao Liu, Xing Han, Na Zhang, and Qiang Liu. Certified monotonic neural networks. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS 2020*, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/b139aeda1c2914e3b579aafd3ceeb1bd-Abstract.html>.
- 33 João Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive Bayes and other linear classifiers with polynomial time and delay. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS 2020*, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/eccd2a86bae4728b38627162ba297828-Abstract.html>.
- 34 João Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska. Explanations for monotonic classifiers. In Marina Meila and Tong Zhang, editors, *ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 7469–7479. PMLR, 2021. URL: <http://proceedings.mlr.press/v139/marques-silva21a.html>.
- 35 Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.
- 36 James B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Math. Program.*, 118(2):237–251, 2009. doi:10.1007/s10107-007-0189-2.
- 37 Manon Ruffini, Jelena Vucinic, Simon de Givry, George Katsirelos, Sophie Barbe, and Thomas Schiex. Guaranteed diversity & quality for the weighted CSP. In *ICTAI 2019*, pages 18–25. IEEE, 2019. doi:10.1109/ICTAI.2019.00012.
- 38 Ethan L. Schreiber, Richard E. Korf, and Michael D. Moffitt. Optimal multi-way number partitioning. *J. ACM*, 65(4):24:1–24:61, 2018. doi:10.1145/3184400.
- 39 Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In *IJCAI*, pages 5103–5111, 2018. doi:10.24963/ijcai.2018/708.

21:18 On the Tractability of Explaining Decisions of Classifiers

- 40 Andy Shih, Arthur Choi, and Adnan Darwiche. Compiling bayesian network classifiers into decision graphs. In *AAAI*, pages 7966–7974, 2019. doi:10.1609/aaai.v33i01.33017966.
- 41 Johan Thapper and Stanislav Zivny. The complexity of finite-valued CSPs. *J. ACM*, 63(4):37:1–37:33, 2016.
- 42 Donald M. Topkis. Minimizing a submodular function on a lattice. *Oper. Res.*, 26(2):305–321, 1978. doi:10.1287/opre.26.2.305.
- 43 Christopher Umans. The minimum equivalent DNF problem and shortest implicants. *J. Comput. Syst. Sci.*, 63(4):597–611, 2001. doi:10.1006/jcss.2001.1775.
- 44 Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *FOCS*, pages 331–342, 2017. doi:10.1109/FOCS.2017.38.