Journal of Artificial Intelligence Research 81 (2024) 401-441 Submitted 07/2023; published 10/2024

A Fortiori Case-Based Reasoning: From Theory to Data

Wijnand van Woerkom W.K.VANWOERKOM@UU.NL
Department of Information and Computing Sciences
Utrecht University, The Netherlands

Davide Grossi D.GROSSI@RUG.NL
Bernoulli Institute for Maths, CS and AI
University of Groningen, The Netherlands
Institute for Logic, Language and Computation
Amsterdam Center for Law and Economics
University of Amsterdam, The Netherlands

Henry Prakken H.PRAKKEN@UU.NL
Department of Information and Computing Sciences
Utrecht University, The Netherlands

Bart Verheij BART.VERHEIJ@RUG.NL
Bernoulli Institute for Maths, CS and AI
University of Groningen, The Netherlands

Abstract

The widespread application of uninterpretable machine learning systems for sensitive purposes has spurred research into elucidating the decision-making process of these systems. These efforts have their background in many different disciplines, one of which is the field of AI & law. In particular, recent works have observed that machine learning training data can be interpreted as legal cases. Under this interpretation, the formalism developed to study case law, called the theory of precedential constraint, can be used to analyze the way in which machine learning systems draw on training data—or should draw on them—to make decisions. In the present work, we advance the theory underlying these explanation methods, by relating it to order theory and logic. This allows us to write a software implementation of the theory that can be used to compute with the definitions and give automatic proofs of the properties of the model. We use this implementation to evaluate the model on a series of datasets. Through this analysis, we characterize the types of datasets that are more, or less, suitable to be described by the theory.

1. Introduction

Much present-day research is focused on making artificial intelligence (AI) more transparent. This work is partially done in response to mounting concerns that uninterpretable algorithms, so-called 'black box' AI, are making high-impact decisions—such as those with legal, social, or ethical consequences—in an unfair or irresponsible manner. A prominent example of such a system is the proprietary software Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), developed by Northepointe, Inc. for automatic risk assessment of various forms of recidivism, which has seen nationwide use in the United States (Angwin et al., 2016). Allegations by Angwin et al. (2016) that COMPAS racially discriminates in its decision-making process have led to a host of follow-up research and

discussions. The COMPAS developers have published a response (Dieterich et al., 2016), and others have pointed to flaws in the original analysis by ProPublica (Barenstein, 2019; Rudin et al., 2020); but as Rudin et al. (2020) point out, this situation is symptomatic of the larger problem that the use of such black box systems is obstructing independent assessment of bias, regardless of the veracity of the allegations in this particular instance.

Many different kinds of solutions have been proposed, among which those to make AI inherently more transparent (Rudin, 2019); to formulate appropriate regulations (Wachter et al., 2017); and to monitor the systems and measure bias over time (Kurita et al., 2019). The line on which the present work builds is that of *post hoc* explainability methods, in which the black-box system is analyzed after it has been trained and little to no access to the way it functions is assumed.

There are in turn many types of post hoc explanation methods, see e.g. the work by Koh and Liang (2017), Ribeiro et al. (2016), and Wachter et al. (2018). We will focus on a particular branch originating from the intersection of AI & law, based on *case-based reasoning* (CBR). The idea of a CBR explanation of a decision is to provide an analogy between it and relevant training examples. Proponents of this approach, such as Nugent and Cunningham (2005), argue that explanations of this form are natural to humans: they are simple, we are well acquainted with reasoning by analogy, and they draw on real evidence in the sense that training examples typically serve as a gold standard that the black box adheres to. Two recent examples of this approach from AI & law are found in the works by Čyras et al. (2016) and Prakken and Ratsma (2022).

The method of Prakken and Ratsma (2022) is based on a formal theory of *precedential constraint*, introduced by Horty (2011), which is a formal framework developed to describe the a fortiori reasoning process underlying case law, i.e., to describe the extent to which a body of precedents constrains a decision in a new case. The key idea of Prakken and Ratsma (2022) behind applying this theory is that the training data used by most modern machine learning systems for binary classification—which consists of rows of data for a set of features together with a binary target variable—can be interpreted as the fact situations of legal cases together with their verdicts. On the basis of this 'training examples as cases' interpretation, Prakken and Ratsma (2022) use the theory of precedential constraint as the theoretical foundation for building a post hoc explanation algorithm. Since the work by Prakken and Ratsma (2022) several other works have appeared that use this interpretation; such as that by Peters et al. (2022, 2023), for developing post-hoc XAI methods; and by Odekerken et al. (2023), who use the model as a classifier for human-in-the-loop decision support.

The overarching goal of the present work is to investigate the extent to which the 'training examples as cases' interpretation is applicable in practice. We do so in three steps. First, we further develop the theory of precedential constraint by connecting it to order theory and formal (many-sorted) logic. Secondly, we use this connection with logic to write an implementation for computing with the theory of precedential constraint using the Z3 *satisfiability modulo theories* (SMT) solver (de Moura & Bjørner, 2008). For example, this implementation allows us to check whether a case base forces the outcome of a novel fact situation, whether a case base is consistent, and whether a given case is a *landmark* (a notion that will be introduced in Section 2.1.4). Thirdly, we use this implementation to analyze various datasets and evaluate the extent to which the data obeys the precedent set by other examples.

For the data analysis, we first compare the output of our implementation with that of the previous results found by Prakken and Ratsma (2022). Then, we instantiate the a fortiori model of precedential constraint on the COMPAS dataset published by Angwin et al. (2016) and subsequently evaluate various statistics. This data is of interest to us for two reasons. First of all, it concerns real-world data rather than fictional data. Secondly, it is highly relevant to the concerns that drive explainable AI research, as automated decisions may be made on the basis of such data which have a big social impact. As such, it is representative of the situations to which our explanation methods may be applied. For the evaluation, we are interested in the *consistency* percentage, which can be thought of as the degree to which the data obeys the precedent set by other examples. Through this analysis we find that an important role is played by what we shall refer to as landmark cases; those cases that set a new precedent with respect to the other cases. We find that in the case of the COMPAS data, a relatively small number of these landmarks force the decision of almost all other cases. Lastly, we instantiate the model on several datasets recently used by Steging et al. (2021). These datasets are mostly of a synthetic nature and have known ground truth labels expressed by logical formulas. This allows us to thoroughly analyze the degree to which the a fortiori model fits these datasets, and makes full use of the capabilities of our implementation which Z3 affords it.

The results of this work can roughly be divided into two parts. In the first part, in Section 2, we fit the theory of precedential constraint, developed by Horty (2011), in the framework of order theory and (many-sorted) logic. We begin this section by explaining the model in Section 2.1. We then give an order-theoretic perspective on this theory in Section 2.2. Then, in Section 2.3, we give a logical perspective on the theory in the language of many-sorted logic, which builds upon the order-theoretic perspective. As the last of our theoretical considerations, we describe under what circumstances the a fortiori model constitutes a binary classifier in Section 2.4.

In the second part of this work, in Section 3, we first show in Section 3.1 that the logical interpretation of the model lets us build an implementation of the theory in Python using the SMT solver Z3. We subsequently put this implementation to work, by instantiating it on a number of datasets and evaluating various statistics. To start, in Section 3.2, we reproduce the results found by Prakken and Ratsma (2022) for the same model. Then, in Section 3.3, we fit the model to real-world recidivism data, which represents the intended domain of XAI applications based on the a fortiori model. In Section 3.4 we look at datasets used by Steging et al. (2021), which allow us to interpret the fit of the a fortiori model to the data on a logical level. We conclude in Section 4 with a summary and discussion of our results, together with some thoughts on future work.

2. The Theory of Precedential Constraint

Horty (2011, 2019) has introduced a framework for formally describing the a fortiori reasoning process underlying case law. This is the type of law arising from the rules or principles used in deciding previous cases called precedents. The underlying idea is that courts must decide similar cases in a similar way. In this way, any particular decision constrains the court in deciding future cases.

In fact, Horty (2011) proposed two formal models of precedential constraint: the *result* model and the *reason* model. In the present work, we focus on the result model, as the contemporary research using the 'training examples as cases' interpretation is based on this version. This is because training examples are readily interpreted as cases in the sense of the result model, while this is not so for the reason model. However, due to the similarity between the two models, our theoretical findings regarding the result model will also be relevant for the reason model.

2.1 A Model of a Fortiori Reasoning

In this section we start out by describing the result model in the set theoretic language used by Horty (2011), illustrate it through some examples, and add the notions of completeness and of landmark cases to the theory.

2.1.1 Dimensions, Preferences, and Cases

In order to describe the fact situation of a case we use what are called dimensions in the AI & law literature, which are formally just partially ordered sets.

Definition 1. A *partial order* $\leq$ on a set P is a relation satisfying the properties:

(1) $a \leq a$ for all $a \in P$;

(2) if $a \leq b$ and $b \leq c$ then $a \leq c$ for all $a, b, c \in P$;

(3) if $a \leq b$ and $b \leq a$ then $a = b$ for all $a, b \in P$.

These properties are respectively called *reflexivity*, *transitivity*, and *antisymmetry*. We say that a partial order is *total*, or *linear*, if for all $a, b \in P$ we have that $a \leq b$ or $b \leq a$.

A *dimension* is a set d together with a partial order $\preceq$ on d, and we assume that there is a finite set D of dimensions $\{d_1, \ldots, d_n\}$ which together describe the relevant aspects of the domain. The idea is that we now specify a fact situation F as we would a point in space—by specifying its value (or coordinate) $F(d)$ in each dimension d. More specifically, a *fact situation* is a choice function on $\{d_1, \ldots, d_n\}$; i.e., a function $F : D \to \bigcup D$ such that $F(d_i) \in d_i$ for $1 \leq i \leq n$. A fact situation can be decided for either of two *outcomes*, also called *sides*, 0 or 1. We will denote an unspecified side with the variable s, and its *opposite* outcome by $\bar{s} := 1 - s$. A *case* (F, s) is now a fact situation F paired with an outcome s. A finite set of cases $\mathcal{C}$ is called a *case base*.

The order $\preceq$ of a dimension d specifies the relative preference the elements of d have towards a particular outcome. More specifically, if $v \preceq w$ for $v, w \in d$ then this means w favors outcome 1 at least as strongly as v, and conversely v prefers outcome 0 at least as strongly as w. Usually we want to compare preference towards an arbitrary outcome s, so to do this we define for any dimension $(d, \preceq)$ the notation $\preceq_s := \preceq$ if $s = 1$ and $\preceq_s := \succeq$ if $s = 0$. Note that by definition we have $\preceq_s = \succeq_{\bar{s}}$.

Example 1. To give some intuition for these definitions we consider a running example of recidivism data. Convicts are described along two dimensions: their age $(d_{\texttt{Age}}, \preceq_{\texttt{Age}})$, and their number of prior offenses $(d_{\texttt{Priors}}, \preceq_{\texttt{Priors}})$. Both of these dimensions have natural

numbers as possible values, so we put $d_{\texttt{Age}} := \mathbb{N}$ and $d_{\texttt{Priors}} := \mathbb{N}$. The outcome for this domain is a judgment of whether the person is at a high (1) or low (0) risk of recidivism. The associated orders are as follows: for age it is the 'greater-than' order $\geq$ on the natural numbers, and for the number of priors we take the 'less-than' order $\leq$. In short, we have

$$(d_{\texttt{Age}}, \preceq_{\texttt{Age}}) := (\mathbb{N}, \geq), \qquad (d_{\texttt{Priors}}, \preceq_{\texttt{Priors}}) := (\mathbb{N}, \leq).$$

The orders we assign to these dimensions represent an assumption that the younger an individual is, and the more priors they have, the more likely they are to recidivate. While this is just an example, these orders are consistent with the literature on recidivism. Beware of the confusion that can arise when the order is the opposite of what is suggested by its symbol; for instance, in our example, we have $40 \preceq_{\texttt{Age}} 20$ because $40 \geq 20$.

2.1.2 The Strength Order

The principle of *stare decisis* states that similar cases must be decided similarly, and so the outcome of any particular case will set a precedent that future decision-making should abide by. Put differently, precedent constrains future decision-making. This is the phenomenon that the theory of precedential constraint tries to model, and it does so principally through usage of the *strength* order, introduced by Horty (2019, Definition 12).

Definition 2. Given fact situations G and F we say F is *at least as strong* as G for an outcome s, denoted $G \preceq_s F$, if it is at least as strong for s on every dimension d:

$$G \preceq_s F \quad \text{if and only if} \quad G(d) \preceq_s F(d) \text{ for all } d \in D.$$

If in addition $(G, s) \in \mathcal{C}$ is a previously decided case in a case base $\mathcal{C}$ then we say that $\mathcal{C}$ *forces* the decision of F for s, and write $\mathcal{C}, F \vDash s$.

This relation models a fortiori reasoning: once a fact situation G was deemed to be sufficiently strong for a side s, any subsequently encountered fact situation F which is even stronger for s—as determined by the strength order $G \preceq_s F$—should also be decided for s.

Example 2. To illustrate this we consider our running example. Suppose that we have classified someone who is 30 years of age and has 5 prior offenses as being at high risk of recidivism. This classification is formalized as a case $(G, 1)$ in a case base $\mathcal{C}$ where the fact situation G given by $G(\texttt{Age}) = 30$ and $G(\texttt{Priors}) = 5$. This decision now dictates that, say, a 24-year-old male with 10 prior offenses should be classified high-risk as well—according to the way we ordered these dimensions. Formally, given a new fact situation F with $F(\texttt{Age}) = 24$, $F(\texttt{Priors}) = 10$, we have $G(\texttt{Age}) \preceq_{\texttt{Age}} F(\texttt{Age})$ as $30 \geq 24$, and $G(\texttt{Priors}) \preceq_{\texttt{Priors}} F(\texttt{Priors})$ as $5 \leq 10$. This is to say that $G \preceq F$, and so $\mathcal{C}, F \vDash 1$ because $(G, 1) \in \mathcal{C}$.

Remark 1. Note that the result model contains an independence assumption between dimensions: if $F(d) \preceq G(d)$ holds for fact situations F and G, then G is considered stronger for the plaintiff along dimension d, regardless of its values in other dimensions. This need not always hold in practice—Prakken and Sartor (1998) give the following example: "[...] even if rain and heat are individually reasons not to go jogging, then the combination of these two factors might very well be instead a reason to go jogging." In situations where this assumption is violated the result model may incorrectly impose constraint.

We mention, as an aside, that this form of reasoning is monotonic in the addition of new cases to the case base, which is formalized by the following simple proposition.

Proposition 1 (Monotonicity)**.** *Let $\mathcal{C}$ and $\mathcal{D}$ be case bases such that $\mathcal{C} \subseteq \mathcal{D}$, F a fact situation, and s an outcome; then $\mathcal{C}, F \vDash s$ implies $\mathcal{D}, F \vDash s$.*

Proof. If $\mathcal{C}, F \vDash s$ then there is some $(G, s) \in \mathcal{C} \subseteq \mathcal{D}$ such that $G \preceq_s F$, so $\mathcal{D}, F \vDash s$. □

Intuitively, monotonicity holds because the newly added cases cannot interfere with previous inferences. In a recently developed version of the result model, which allows for multi-step inferences, the monotonicity property fails (van Woerkom et al., 2023).

2.1.3 Consistency and Completeness

In addition to making decisions about new fact situations on the basis of a case base and the strength order, we can consider the degree to which the cases within a case base are consistent with each other, relative to the strength order. For instance, if a case base $\mathcal{C}$ contains cases (F, s) and (G, t) such that $F \preceq_s G$ then there are two possibilities: either $s = t$, in which case the decision of G for t is consistent with that of F for s; or $s \neq t$, in which case the decision of G for t is inconsistent with that of F for s.

Definition 3. A case (F, s) is said to be *inconsistent* with respect to a case base $\mathcal{C}$ when $\mathcal{C}, F \vDash \bar{s}$, and *consistent* otherwise. A case base is said to be consistent when all of its cases are, and inconsistent otherwise.

Remark 2. Note that the presence of an inconsistent case with outcome s implies the presence of an inconsistent case with outcome $\bar{s}$. To see why, consider an inconsistent case $(F, s) \in \mathcal{C}$; so $\mathcal{C}, F \vDash \bar{s}$. This means there is a case $(G, \bar{s}) \in \mathcal{C}$ such that $G \preceq_{\bar{s}} F$. But then, by definition, $F \preceq_s G$ and so $\mathcal{C}, G \vDash s$; which is to say that $(G, \bar{s})$ is inconsistent. In practice, this means that in order to check whether a case base is consistent, it suffices to check whether all of its cases with 1 specific outcome are consistent (which may save work).

Consistency, as thus defined, is a binary property—a case base is either consistent or it is not. It can be made a quantitative property by considering the relative frequency of consistent cases in the case base, and we will do so in our experiments to come in Section 3.

Now let us consider the following alternative formulation of consistency.

Lemma 1. *A case base is consistent iff there is no fact situation with both outcomes forced.*

Proof. Suppose that a case base is inconsistent according to Definition 3; this means there is a case $(F, s) \in \mathcal{C}$ with $s \in \{0, 1\}$ such that $\mathcal{C}, F \vDash \bar{s}$. Note that for any case $(F, s) \in \mathcal{C}$ we have $\mathcal{C}, F \vDash s$ since $F \preceq_s F$, and so we have found a fact situation F for which both $\mathcal{C}, F \vDash 0$ and $\mathcal{C}, F \vDash 1$. Conversely, suppose there is a fact situation F such that both $\mathcal{C}, F \vDash 0$ and $\mathcal{C}, F \vDash 1$. Then there is $(G, 0), (H, 1) \in \mathcal{C}$ such that $G \preceq_0 F$ and $H \preceq_1 F$. But then, by definition, $F \preceq_1 G$ and so $H \preceq_1 G$ by transitivity; this means $(G, 0)$ is inconsistent. □

We see that consistency states that there is no fact situation for which both outcomes are forced by the case base. This property has a natural counterpart which—to the best of our knowledge—has not yet appeared in the literature, and which is defined as follows.

Table 1: An example case base for the `Age` and `Priors` dimensions, which is neither consistent nor complete. See Figure 1 for a graphical representation. The second, third, and fifth row correspond respectively to the fact situations F, H, and G from Example 3.

	`Age`	`Priors`	**Label**
	30	1	0
F	35	5	0
	45	4	0
	30	2	1
	35	7	1
G	40	3	1
H	45	7	–

Definition 4. A case base is *complete* when every fact situation has an outcome forced.

Remark 3. The terminology we propose in Definition 4 is inspired by the notion of completeness of a set of formulas in logic: a set of formulas T is *consistent* if there is no formula ϕ such that both $T \vDash \phi$ and $T \vDash \neg\phi$, and it is *complete* if for all formulas ϕ either $T \vDash \phi$ or $T \vDash \neg\phi$ (Bradley & Manna, 2007, Section 3.1). Compare this with our definitions here: a case base $\mathcal{C}$ is *consistent* if there is no fact situation F such that both $\mathcal{C}, F \vDash 0$ and $\mathcal{C}, F \vDash 1$, and it is *complete* if for all fact situations F either $\mathcal{C}, F \vDash 0$ or $\mathcal{C}, F \vDash 1$. The similarity is somewhat superficial, though, as there are important differences between these notions; for instance, an inconsistent set of logical formulas is necessarily complete by the *ex falso* principle, while for case bases this implication does not hold.

Example 3. An example case base for our recidivism example can be found in Table 1. This case base is neither consistent nor complete. It is not consistent because the case $(F, 0)$ with $F(\texttt{Age}) = 35$ and $F(\texttt{Priors}) = 5$ has its outcome forced for 1 by the case $(G, 1)$ with $G(\texttt{Age}) = 40$ and $G(\texttt{Priors}) = 3$. It is not complete because there are (infinitely many) fact situations that do not have their outcome forced for either 0 or 1, such as the listed fact situation H with $H(\texttt{Age}) = 45$ and $H(\texttt{Priors}) = 7$.

A case base might become complete, or inconsistent, through the addition of new cases. Conversely, a case base can be made incomplete, or consistent, through the removal of cases. Any set of dimensions D trivially admits a sound case base; namely the empty case base $\emptyset$. It would also trivially admit a complete case base if not for our requirement that case bases are finite: simply decide all fact situations for outcome 0, or 1, or any mix thereof. Since the choice of dimensions D may give rise to an infinite number of fact situations, this may be impossible, and in fact it is impossible for our running example—as we now show.

Proposition 2. *There is no complete case base for the* `Age` *and* `Priors` *dimensions.*

Proof. Let $\mathcal{C}$ be any case base for the `Age` and `Priors` dimensions; we prove the proposition by constructing a fact situation F which does not have its outcome forced by $\mathcal{C}$:

$$F(\texttt{Age}) = 1 + \max_{(G,1)\in\mathcal{C}} G(\texttt{Age}), \qquad F(\texttt{Priors}) = 1 + \max_{(G,0)\in\mathcal{C}} G(\texttt{Priors}).$$

This fact situation F exists because the case base $\mathcal{C}$ is finite. The claim is that F does not have its outcome forced by $\mathcal{C}$. Suppose, to the contrary, that F were forced for 0; then there is a case $(H, 0) \in \mathcal{C}$ such that $F \preceq H$. This means that $F(\texttt{Priors}) \preceq H(\texttt{Priors})$, and so $F(\texttt{Priors}) = 1 + \max_{(G,0)\in\mathcal{C}} G(\texttt{Priors}) \leq H(\texttt{Priors})$ (note the order used here). This implies that $\max_{(G,0)\in\mathcal{C}} G(\texttt{Priors}) < H(\texttt{Priors})$, contradicting $(H, 0) \in \mathcal{C}$. If F had its outcome forced for 1 then a similar contradiction occurs with the definition of $F(\texttt{Age})$. □

This proposition demonstrates that there are sets of dimensions D which—from the onset—do not admit any complete case base, because of the requirement that case bases are finite. We maintain this requirement because real-world datasets are necessarily finite, and because it allows us in general to construct logical formulas describing the forcing behavior of the case base, as we will show in Section 2.3.

2.1.4 Landmark Cases

Of particular interest with respect to the forcing relation are what we call landmark cases. The motivating idea is that when a case has its outcome forced by another, it is—by transitivity of the strength order—rendered superfluous as a precedent. As such, the most salient cases are those that do not have their outcome forced by another case.

Definition 5. A case $(F, s) \in \mathcal{C}$ is called a *landmark* case if $\mathcal{C} \setminus \{(F, s)\}, F \nvDash s$. Cases that are not landmarks are called *regular*. We let $\mathcal{L} \subseteq \mathcal{C}$ denote the subset of $\mathcal{C}$ containing just its landmark cases.

Intuitively speaking, a case is a landmark when the decision of its fact situation is not forced for its outcome by the rest of the case base. Do note, however, that a case (F, s) can be a landmark while $\mathcal{C} \setminus \{(F, s)\}, F \vDash \bar{s}$.

The relevance of landmarks is described by the following proposition.

Proposition 3. *For a case base $\mathcal{C}$, a fact situation F, and an outcome s, we have*

$$\mathcal{C}, F \vDash s \iff \mathcal{L}, F \vDash s.$$

The direction from right to left is just an instance of monotonicity, but the other direction is somewhat more difficult to justify. We defer a proof until the next section, as we will develop some notation in the meantime that will ease this task.

2.2 An Order-Theoretic Perspective

The mathematical tools used in Horty's model of a fortiori reasoning have been studied more generally, as part of a branch of mathematics known as order theory: the study of binary relations on sets that correspond intuitively to the notion of order (as in Definition 1). In this section, we recall some notions from order theory and relate them to Horty's model. We do this because these notions help clarify the formal aspect of Horty's model, and because we will make use of them in Section 2.3 where we relate the model to (many-sorted) logic. See the work by Davey and Priestley (2002) for a detailed introduction to order theory and its connection with logic.

2.2.1 The Product Order and its Up- and Down-Sets

Given a set P of sets, the *product* of P, denoted by $\prod P$, is the set containing all choice functions on P;

$$\prod P := \{f : P \to \bigcup P \mid f(A) \in A \text{ for all } A \in P\}.$$

If every set $A \in P$ comes with a partial order $\leq_A$, then P can itself be partially ordered. In fact, this can be done in multiple ways, but we will use what is called the *coordinatewise order* or *product order* $\leq_\prod$, which is defined for $f, g \in \prod P$ by $f \leq_\prod g$ if and only if $f(A) \leq_A g(A)$ for all $A \in P$ (Davey & Priestley, 2002, p. 18).

We have seen a particular instance of this construction in Section 2.1; given a set of dimensions D we let $\mathcal{F} := \prod D$ denote the set of fact situations, and write $\preceq$ for the product order on $\mathcal{F}$, which is known as the strength order in the theory of precedential constraint.

A case base is a finite subset $\mathcal{C} \subseteq \mathcal{F} \times \{0, 1\}$, but we can also think of $\mathcal{C}$ as comprising two designated subsets of $\mathcal{F}$; one $\mathcal{C}_0 \subseteq \mathcal{F}$ containing the fact situations of cases with outcome 0, and one $\mathcal{C}_1 \subseteq \mathcal{F}$ with those that received outcome 1. Given a case base $\mathcal{C}$, we identify these subsets with the notation $\mathcal{C}_s := \{F \in \mathcal{F} \mid (F, s) \in \mathcal{C}\}$.

A concept from order theory that we will use extensively is that of *up-sets* and *down-sets*. Given an ordered set $(P, \leq)$ and a subset $A \subseteq P$, we define its up- and down-sets $\uparrow A, \downarrow A$ by

$$\uparrow A := \{b \in P \mid a \leq b \text{ for some } a \in A\}, \qquad \downarrow A := \{b \in P \mid b \leq a \text{ for some } a \in A\}.$$

If A is a singleton $\{a\}$ we may write $\uparrow a$ instead of $\uparrow\{a\}$; note that $\uparrow A = \bigcup_{a \in A} \uparrow a$. These operations satisfy the conditions of closure operations (Davey & Priestley, 2002, Chapter 7).

Lemma 2. *Let $(P, \leq)$ be a partially ordered set. The upset operation $\uparrow$ on P is a* closure *operation, meaning it satisfies the following properties for all subsets $A, B \subseteq P$:*

- $A \subseteq \uparrow A$;
- *if $A \subseteq B$ then $\uparrow A \subseteq \uparrow B$;*
- $\uparrow\uparrow A = \uparrow A$.

The same holds for the down-set operator $\downarrow$.

Working with fact situations we are also interested in the opposite of the product order, for which we use the notation $\preceq_s := \preceq$ if $s = 1$ and $\preceq_s := \succeq$ if $s = 0$. We will do the same for the up- and down-set notation: $\uparrow_s := \uparrow$ if $s = 1$ and $\uparrow_s := \downarrow$ if $s = 0$.

2.2.2 Forcing as Up- and Down-Set Membership

This concept of up- and down-sets is useful because it is closely related to the definition of case base forcing. The following is just a simple rephrasing of definitions, but we state it explicitly because we will use it frequently.

Lemma 3. *$\mathcal{C}, F \vDash s$ is equivalent to $F \in \uparrow_s \mathcal{C}_s$.*

Proof. We have $\mathcal{C}, F \vDash s \iff G \preceq_s F$ for some $G \in \mathcal{C}_s \iff F \in \uparrow_s \mathcal{C}_s$. □

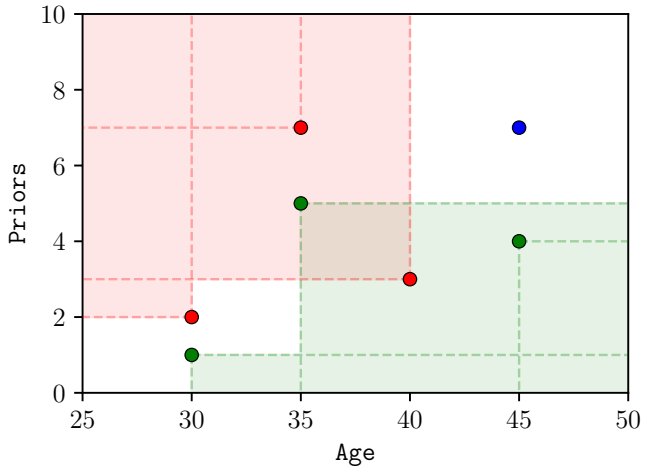


Figure 1: An illustration of the case base in Table 1. Green dots are cases with outcome 0, red dots are cases with outcome 1, and the shaded regions indicate their up- and down-sets in the strength order, which we call forcing cones. The blue dot is a counterexample to completeness.

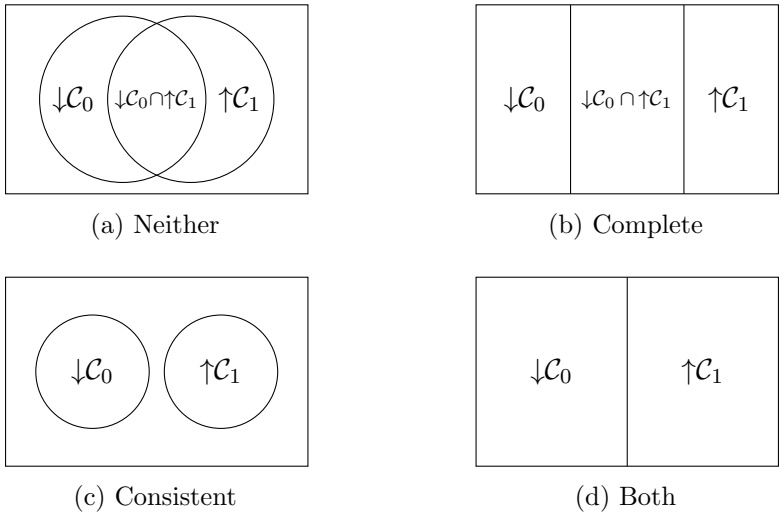
Example 4. We consider again the case base of Example 3, listed in Table 1. The dimensionality of $\mathcal{F} = \mathbb{N} \times \mathbb{N}$ is low and so we can visualize it—see Figure 1. The up-sets of the cases are also shown, which we will call *forcing cones*. For instance, for the case $(F, 1)$ with $F(\texttt{Age}) = 30$ and $F(\texttt{Priors}) = 2$, any fact situation G with $G(\texttt{Age}) \leq 20$ and $G(\texttt{Priors}) \geq 2$ will have greater strength for outcome 1 than F, and so will be forced for side 1.

The visualization in Figure 1 shows that concepts of interest can be phrased in terms of the forcing cones; for instance, we can see landmarks as cases that are not within a forcing cone of a case with its own outcome. Furthermore, inconsistency corresponds to overlapping cones of cases with opposite outcomes, and completeness corresponds to areas that are not covered by any cones. If there are no overlapping forcing cones of different outcomes, then the case base is consistent; likewise, if the whole space of fact situations is covered by the forcing cones, then the case base is complete. This is stated formally by the following lemma.

Lemma 4. *A case base $\mathcal{C}$ is consistent iff $\downarrow\mathcal{C}_0 \cap \uparrow\mathcal{C}_1 = \emptyset$, and complete iff $\downarrow\mathcal{C}_0 \cup \uparrow\mathcal{C}_1 = \mathcal{F}$.*

Remark 4. As mentioned in Remark 2, for consistency it also suffices to check either of the equations $\mathcal{C}_0 \cap \uparrow\mathcal{C}_1 = \emptyset$ or $\mathcal{C}_1 \cap \downarrow\mathcal{C}_0 = \emptyset$.

The visualization of Figure 1 is possible because our example only has two dimensions—with more than two such a visualization becomes impractical. However, in the general case we can still usefully visualize the forcing cones using Euler diagrams. For example, Lemma 4 relates consistency and completeness to the sets $\downarrow\mathcal{C}_0 \cap \uparrow\mathcal{C}_1$ and $\mathcal{F} \setminus (\downarrow\mathcal{C}_0 \cup \uparrow\mathcal{C}_1)$ being empty. So, the four possible situations with regards to the status of consistency and completeness of a case base can be visualized using Euler diagrams; see Figure 2. We will make use of such visualizations for our data analysis in Section 3.



(a) Neither (b) Complete

(c) Consistent (d) Both

Figure 2: Euler diagram representations of the consistency and completeness properties.

2.2.3 Landmarks as Minimal and Maximal Elements

Another useful concept from order theory is that of minimal and maximal elements. Given a partially ordered set $(P, \leq)$ and a subset $A \subseteq P$, we say an element $a \in A$ is *minimal in* A if there is no $b \in A$ such that $b < a$. We denote the set of minimal elements of A by $\min A$. Dually, we say $a \in A$ is *maximal in* A if there is no $b \in A$ such that $a < b$. We denote the set of maximal elements of A by $\max A$. Again, we define some notion to account for the two sides: $\min_s := \min$ if $s = 1$ and $\min_s := \max$ if $s = 0$.

This notion makes it easier to understand the set of landmarks $\mathcal{L}$ of a case base $\mathcal{C}$:

$$\begin{aligned} F \in \mathcal{L}_s &\iff (F, s) \in \mathcal{L} \\ &\iff \mathcal{C} \setminus \{(F, s)\}, F \not\models s \\ &\iff G \not\prec_s F \text{ for all } (G, s) \in \mathcal{C} \\ &\iff F \in \min_s \mathcal{C}_s. \end{aligned}$$

This means that $\mathcal{L}_s = \min_s \mathcal{C}_s$, or more explicitly, that $\mathcal{L}_0 = \max \mathcal{C}_0$ and $\mathcal{L}_1 = \min \mathcal{C}_1$. This is a useful fact in combination with the following (well-known) lemma.

Lemma 5. *If $(P, \leq)$ is a partially ordered set then any finite $A \subseteq P$ satisfies $\uparrow A = \uparrow \min A$.*

Proof. We prove the inclusions separately, using the properties listed in Lemma 2.

($\subseteq$) First we note that since A is finite, there is for every $a_1 \in A$ a finite descending chain $a_1 > a_2 > \cdots > a_n$ within A, so $a_1 \geq a_n$ by transitivity, for some $a_n \in \min A$. This is to say that $A \subseteq \uparrow \min A$, and therefore $\uparrow A \subseteq \uparrow\uparrow \min A = \uparrow \min A$ as desired.

($\supseteq$) It follows from $\min A \subseteq A$ that $\uparrow \min A \subseteq \uparrow A$. □

Corollary 1. *For any case base $\mathcal{C}$ we have $\uparrow_s \mathcal{C}_s = \uparrow_s \mathcal{L}_s$.*

Proof. Immediate from Lemma 5 and the fact that $\mathcal{L}_s = \min_s \mathcal{C}_s$. □

Proposition 3 now also follows immediately from Corollary 1 and Lemma 3.

Proof of Proposition 3. $\mathcal{C}, F \models s \iff F \in \uparrow_s \mathcal{C}_s \iff F \in \uparrow_s \mathcal{L}_s \iff \mathcal{L}, F \models s.$ □

Intuitively, what Proposition 3 tells us is that when a fact situation is forced by a case base then it is also forced by a landmark case of that case base. This means that in order to get an understanding of the behavior of the strength order of a case base, it suffices to consider the landmark cases. This can be a very useful reduction in practice—we will see some datasets that contain thousands of cases but only some handfuls of landmarks.

2.3 A Logical Perspective

In this section, we phrase the a fortiori model from the point of view of logic. To do this we use many-sorted logic, so we begin in Section 2.3.1 by describing this general framework, and then proceed in Section 2.3.2 to demonstrate that the a fortiori model can be phrased as an instance of this framework. There are many works in the literature giving such descriptions; see for example the work by de Moura and Bjørner (2009) and Manzano and Aranda (2022).

We use many-sorted logic, as opposed to e.g. Liu et al. (2022) who use modal logic, as it is the type of logic used in contemporary SMT solvers. This means that the rephrasing in logic allows us to use the machinery of SMT solvers to reason about specific instances of the a fortiori model. We describe how this can be done in Section 3.1, and use this implementation in subsequent sections to evaluate the a fortiori model on several datasets.

2.3.1 Many-Sorted Logic

Many-sorted logic is very similar to unsorted logic—it revolves around questions of satisfiability of formulas built from the familiar logical connectives as well as function and relation symbols from some signature. The difference is that these symbols have some *sort* which can be thought of as a datatype in programming. Examples of such sorts are those corresponding to integers, rationals, or boolean values.

More formally, we assume there is a finite set of sorts $S = \{s_1, \ldots, s_n\}$.[1] A *signature* Σ over S is a set of *function* and *predicate* symbols, together with a map $\text{ar} : \Sigma \to S^+$ where S^+ is the set of n-tuples of elements of S for all $n \geq 1$. The map ar associates each element of Σ with an *arity*. For a function symbol $f \in \Sigma$ we write $f : s_1 \times \ldots \times s_n \to s$ instead of $\text{ar}(f) = (s_1, \ldots, s_n, s)$. If $n = 0$ we say f is a *constant*. Similarly, for a relation symbol $R \in \Sigma$ we write $R : s_1 \times \ldots \times s_n$ instead of $\text{ar}(R) = (s_1, \ldots, s_n)$. In addition, we assume there is a set of *variables* X, each of which is associated with a sort; we write $x : s$ to denote that $x \in X$ is of sort $s \in S$.

Any signature Σ over a set of sorts S induces a set T^Σ of *terms*, each of which again has a specific sort. We write $t : s$ to mean that $t \in T^\Sigma$ is of sort $s \in S$. These terms are inductively defined as the smallest set t^Σ such that:

- any variable $x : s$ is a term of sort s;
- given a function symbol $f : s_1 \times \ldots \times s_n \to s$ of Σ and terms $t_1 : s_1, \ldots, t_n : s_n$ there is a term $f(t_1, \ldots, t_n) : s$ of sort s.

1. The notation s_i for sorts clashes with the notation for case outcomes; we let the context disambiguate.

Formulas of many-sorted logic are built using the terms T^Σ together with the usual logical symbols $\top, \bot, \wedge, \vee, \neg, \to, \leftrightarrow$, and with an equality symbol $\doteq_s : s \times s$ for every sort $s \in S$. In practice, we will usually omit the subscript s and let the context determine which of the equality symbols is used. Using these we build the set L^Σ_{at} of *atomic* Σ*-formulas* with:

- $\bot, \top$;
- $t_1 \doteq_s t_2$ for every sort $s \in S$ and terms $t_1 : s$, $t_2 : s$;
- $R(t_1, \ldots, t_n)$ for every relation symbol $R : s_1 \times \ldots \times s_n$ and terms $t_1 : s_1, \ldots, t_n : s_n$.

From the atomic Σ-formulas we now inductively build the full set of Σ-formulas L^Σ as the smallest set such that:

- $L^\Sigma_{\text{at}} \subseteq L^\Sigma$;
- if $\phi \in L^\Sigma$ then $\neg\phi \in L^\Sigma$;
- if $\phi, \psi \in L^\Sigma$ then $\phi \wedge \psi, \phi \vee \psi, \phi \to \psi, \phi \leftrightarrow \psi \in L^\Sigma$;
- if $\phi \in L^\Sigma$ and $x : s$ is a variable then $(\forall x : s)\phi, (\exists x : s)\phi \in L^\Sigma$.

Furthermore, given an indexed set of formulas $\{\phi_i \mid i \in I\} \subseteq L^\Sigma$ for a finite set I we define:

$$\bigwedge_{i \in I} \phi_i := \phi_{i_1} \wedge \cdots \wedge \phi_{i_n}, \qquad \bigvee_{i \in I} \phi_i := \phi_{i_1} \vee \cdots \vee \phi_{i_n}.$$

The formulas in L^Σ can now be interpreted in *structures* assigning meaning to the sorts and symbols. More specifically, a Σ*-structure* $\mathbb{A} = ((A_s)_{s \in S}, I)$ associates to each sort $s \in S$ a set A_s; to each function symbol $f : s_1 \times \ldots \times s_n \to s \in \Sigma$ a function $f_I : A_{s_1} \times \ldots \times A_{s_n} \to A_s$; and to each relation symbol $R : s_1 \times \ldots \times s_n \in \Sigma$ a relation $R_I \subseteq A_{s_1} \times \ldots \times A_{s_n}$. An *assignment* α is an assignment of meaning to the variables, according to their sort. More specifically, an assignment α is a function on X such that $\alpha(x : s) \in A_s$ for $x : s \in X$. Such an assignment can be extended to operate on the set of terms T^Σ by recursively defining $\alpha(f(t_1, \ldots, t_n) : s) = f_I(\alpha(t_1), \ldots, \alpha(t_n))$.

Given a Σ-structure $\mathbb{A}$ and an assignment α we can now define what it means for a formula $\phi \in L^\Sigma$ to be *true* in $\mathbb{A}$. We do so by induction on the complexity of formulas:

$$\begin{aligned}
\mathbb{A}, \alpha &\vDash \bot \text{ is never true,} \\
\mathbb{A}, \alpha &\vDash \top \text{ is always true,} \\
\mathbb{A}, \alpha \vDash t_1 \doteq t_2 &\iff \alpha(t_1) = \alpha(t_2), \\
\mathbb{A}, \alpha \vDash R(t_1, \ldots, t_n) &\iff R_I(\alpha(t_1), \ldots, \alpha(t_n)), \\
\mathbb{A}, \alpha \vDash \neg\phi &\iff \mathbb{A}, \alpha \nvDash \phi, \\
\mathbb{A}, \alpha \vDash \phi \wedge \psi &\iff \mathbb{A}, \alpha \vDash \phi \text{ and } \mathbb{A}, \alpha \vDash \psi, \\
\mathbb{A}, \alpha \vDash \phi \vee \psi &\iff \mathbb{A}, \alpha \vDash \phi \text{ or } \mathbb{A}, \alpha \vDash \psi, \\
\mathbb{A}, \alpha \vDash \phi \to \psi &\iff \text{if } \mathbb{A}, \alpha \vDash \phi \text{ then } \mathbb{A}, \alpha \vDash \psi, \\
\mathbb{A}, \alpha \vDash \phi \leftrightarrow \psi &\iff \mathbb{A}, \alpha \vDash \phi \text{ if and only if } \mathbb{A}, \alpha \vDash \psi.
\end{aligned}$$

Given some set $T \subseteq L^\Sigma$, often called a *theory*, we write $\mathbb{A}, \alpha \vDash T$ if $\mathbb{A}, \alpha \vDash \phi$ for every $\phi \in T$.

A notion of central importance in logic is satisfiability. A formula ϕ is said to be *satisfiable* if there exists a model $\mathbb{A}$ and an assignment α such that $\mathbb{A}, \alpha \vDash \phi$, and *unsatisfiable* otherwise. Computer scientists—as opposed to model theorists—are often interested in a more restricted notion of satisfiability, which fixes the structure $\mathbb{A}$. For this reason, we may also say a formula is $\mathbb{A}$-*satisfiable* if there is some assignment α such that $\mathbb{A}, \alpha \vDash \phi$. Often we are particularly interested in the satisfiability of a formula ϕ relative to some set of background restrictions given by a theory T. To this end, we say a formula is $\mathbb{A}$-satisfiable *modulo a theory* T, denoted $\mathbb{A}, \alpha \vDash_T \phi$, if $\mathbb{A}, \alpha \vDash T \cup \{\phi\}$.

In the remainder of this work, we will be working with satisfiability for some fixed structure $\mathbb{A}$, modulo some background theory T. Therefore, in order not to clutter notation, we will simply speak of satisfiability when we mean $\mathbb{A}$-satisfiability modulo T, and write $\mathbb{A}, \alpha \vDash \phi$ when we mean $\mathbb{A}, \alpha \vDash_T \phi$. The particular structure $\mathbb{A}$ and background theory T relative to which we are referring to will either be irrelevant or clear from the context.

We conclude this section with some notions closely related to satisfiability. Two formulas $\phi, \psi \in L^\Sigma$ are said to be *equivalent*, denoted $\phi \equiv \psi$, when for all assignments α we have $\mathbb{A}, \alpha \vDash \phi$ if and only if $\mathbb{A}, \alpha \vDash \psi$. A formula ϕ is *valid* if $\mathbb{A}, \alpha \vDash \phi$ for all assignments α. We define the *semantics* $[\![\phi]\!]$ of a formula ϕ as the set of all satisfying assignments α; i.e. $[\![\phi]\!] := \{\alpha \mid \mathbb{A}, \alpha \vDash \phi\}$. These notions are all related, as the following equivalences show:

$$\begin{aligned} \phi \text{ is valid} &\iff \neg\phi \text{ is unsatisfiable} \\ &\iff \phi \equiv \top, \\ \phi \text{ is unsatisfiable} &\iff \phi \equiv \bot, \\ \phi \leftrightarrow \psi \text{ is valid} &\iff \phi \equiv \psi \\ &\iff [\![\phi]\!] = [\![\psi]\!]. \end{aligned}$$

2.3.2 A Logical Formulation of the a Fortiori Model

We now show precedential constraint can be framed in terms of a many-sorted logic. For a given set of dimensions D, we take D as the set of sorts and define a signature $\Sigma(D)$ by

$$\Sigma(D) := \{c_v \mid v \in d \in D\} \cup \{\sqsubseteq_d \mid d \in D\}.$$

In other words, we introduce for every dimension $d \in D$ the following set of symbols: a constant c_v with $c_v : d$ for every value $v \in d$, and a relation symbol $\sqsubseteq_d$ with $\sqsubseteq_d : d \times d$ for the dimension order of d. For the variables we take a set X with precisely one variable x_d with $x_d : d$ for each dimension d; so $X := \{x_d \mid d \in D\}$.

We now fix a structure $\mathbb{D} = (D, I)$ for this signature: the domains are given by the set of dimensions D, and the interpretation I simply interprets the symbols according to their intended meaning: $I(c_v) := v$ and $I(\sqsubseteq_d) := \preceq_d$. Due to this fixed interpretation—and to avoid notational clutter—we may henceforth, by abuse of notation, write v where we mean c_v and $\preceq$ where we mean $\sqsubseteq_d$. So for instance, for $v, w \in d$ we will simply write $v \preceq w$ where, strictly speaking, we should write $c_w \sqsubseteq_d c_v$. We will also use the subscript s notation as we did before.

Now, an assignment α of this language is a function on X that maps a variable to a value of the type of that variable, i.e. $\alpha(x_d) \in d$. Since the variables correspond one-to-one

with the dimensions, this means that an assignment for this language is essentially the same thing as a fact situation. Therefore, we will henceforth treat the two as interchangeable and use $F, G, H, \ldots$ as variables to denote assignments.

Lastly, we need a background theory T relative to which we phrase satisfiability. For instance, T should specify precisely how the elements of the dimensions are related to each other in their respective orders or other axioms related to the dimensions. For instance, for the $\texttt{Age}$ dimension $(\mathbb{N}, \geq)$ we need a theory T that includes the theory of the natural numbers, in order to interpret, e.g., the constants that will appear in the formulas. Bradley and Manna (2007, Section 3) and Bjørner and Nachmanson (2020) give many examples of such theories.

Example 5. We consider the structure and language for our running example with dimensions $(\texttt{Age}, \preceq_{\texttt{Age}}) = (\mathbb{N}, \geq)$ and $(\texttt{Priors}, \preceq_{\texttt{Priors}}) = (\mathbb{N}, \leq)$. For some fact situation F we can now form a formula $x_{\texttt{Age}} \sqsubseteq_{\texttt{Age}} c_{F(\texttt{Age})} \wedge c_{F(\texttt{Priors})} \sqsubseteq_{\texttt{Priors}} x_{\texttt{Priors}}$. What does it mean for this formula to be satisfiable? Let G be any assignment, then

$$\begin{aligned}
&\mathbb{D}, G \vDash x_{\texttt{Age}} \sqsubseteq_{\texttt{Age}} c_{F(\texttt{Age})} \wedge c_{F(\texttt{Priors})} \sqsubseteq_{\texttt{Priors}} x_{\texttt{Priors}} \\
&\iff \mathbb{D}, G \vDash x_{\texttt{Age}} \sqsubseteq_{\texttt{Age}} c_{F(\texttt{Age})} \text{ and } \mathbb{D}, G \vDash c_{F(\texttt{Priors})} \sqsubseteq_{\texttt{Priors}} x_{\texttt{Priors}} \\
&\iff I(\sqsubseteq_{\texttt{Age}})(G(x_{\texttt{Age}}), c_{F(\texttt{Age})}) \text{ and } I(\sqsubseteq_{\texttt{Priors}})(c_{F(\texttt{Priors})}, x_{\texttt{Priors}}) \\
&\iff G(x_{\texttt{Age}}) \geq F(\texttt{Age}) \text{ and } F(\texttt{Priors}) \leq G(x_{\texttt{Priors}}) \\
&\iff F \preceq G.
\end{aligned}$$

So, an assignment G satisfies this formula if and only if $F \preceq G$. In other words,

$$[\![x_{\texttt{Age}} \sqsubseteq_{\texttt{Age}} c_{F(\texttt{Age})} \wedge c_{F(\texttt{Priors})} \sqsubseteq_{\texttt{Priors}} x_{\texttt{Priors}}]\!] = {\uparrow}F.$$

Example 5 shows that the semantics of formulas in $L^{\Sigma(D)}$ can correspond to subsets of interest. In general, since assignments correspond to fact situations, the semantics function associates each formula $\phi \in L^{\Sigma(D)}$ to some subset $[\![\phi]\!] \subseteq \mathcal{F}$ of satisfying assignments.

Lemma 6. *The following equations hold for the semantics function* $[\![-]\!] : L^{\Sigma(D)} \to P(\mathcal{F})$*;*

$$\begin{aligned}
[\![\bot]\!] &= \emptyset, \\
[\![\top]\!] &= \mathcal{F}, \\
[\![v \doteq x_d]\!] &= \{F \in \mathcal{F} \mid v = F(d)\} \\
[\![v \preceq_s x_d]\!] &= \{F \in \mathcal{F} \mid v \preceq_s F(d)\}, \\
[\![\neg\phi]\!] &= \mathcal{F} \setminus [\![\phi]\!], \\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!], \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!].
\end{aligned}$$

Proof. By a routine induction on the complexity of formulas in $L^{\Sigma(D)}$. □

Using the semantics function $[\![-]\!]$ we can now generalize Example 5 and show that questions related to the a fortiori model can be phrased in terms of formula satisfiability in the structure $\mathbb{D}$. In particular, we will relate the notion of forcing, landmarks, consistency, and completeness to satisfiability in $\mathbb{D}$.

To start, let F be a fact situation; we define a formula $\phi_s(F) \in L^{\Sigma(D)}$ which states that the variable fact situation x is at least as strong for side s as F:

$$\phi_s(F) := \bigwedge_{d \in D} F(d) \preceq_s x_d. \tag{1}$$

Using the equations of Lemma 6 we can now easily derive that $[\![\phi_s(F)]\!] = \uparrow_s F$:

$$\begin{aligned}
[\![\phi_s(F)]\!] &= \left[\!\!\left[\bigwedge_{d \in D} F(d) \preceq_s x_d\right]\!\!\right] \\
&= \bigcap_{d \in D} [\![F(d) \preceq_s x_d]\!] \\
&= \bigcap_{d \in D} \{G \in \mathcal{F} \mid F(d) \preceq_s G(d)\} \\
&= \{G \in \mathcal{F} \mid F \preceq_s G\} \\
&= \uparrow_s F.
\end{aligned}$$

Next, using $\phi_s(F)$, we define a formula $\Phi_s(\mathcal{C}) \in L^{\Sigma(D)}$ which states that the variable fact situation x has its outcome forced for s by the case base $\mathcal{C}$:

$$\Phi_s(\mathcal{C}) := \bigvee_{F \in \mathcal{C}_s} \phi_s(F). \tag{2}$$

Using the semantics of $\phi_s(F)$, it is easy to see that $[\![\Phi_s(\mathcal{C})]\!] = \uparrow_s \mathcal{C}_s$:

$$[\![\Phi_s(\mathcal{C})]\!] = \left[\!\!\left[\bigvee_{F \in \mathcal{C}_s} \phi_s(F)\right]\!\!\right] = \bigcup_{F \in \mathcal{C}_s} [\![\phi_s(F)]\!] = \bigcup_{F \in \mathcal{C}_s} \uparrow_s F = \uparrow_s \mathcal{C}_s.$$

Using this we can relate the notion of satisfiability of $\Phi_s(\mathcal{C})$ and the forcing relation induced by the case base $\mathcal{C}, F \vDash s$. This result formally establishes the connection between the a fortiori model and the reformulation we present in many-sorted logic.

Proposition 4. *$\mathbb{D}, F \vDash \Phi_s(\mathcal{C})$ if and only if $\mathcal{C}, F \vDash s$.*

Proof. Combining the previous results we get:

$$\mathbb{D}, F \vDash \Phi_s \iff F \in [\![\Phi_s(\mathcal{C})]\!] \iff F \in \uparrow_s \mathcal{C}_s \iff \mathcal{C}, F \vDash s. \qquad \square$$

To make claims about a particular fact situation we need a way to fix the interpretation. There is no symbol in our language for directly equating a fact situation F to the variable fact situation x (i.e. $F \doteq x$ is not a valid formula), but we can define a formula $F \mathrel{\mathring{=}} x$ amounting to the same:

$$F \mathrel{\mathring{=}} x := \bigwedge_{d \in D} F(d) \doteq x_d. \tag{3}$$

Again, we can use the equations of Lemma 6 to show that this has the intended semantics:

$$\begin{aligned}
[\![F \mathrel{\mathring{=}} x]\!] &= \left[\!\!\left[\bigwedge_{d \in D} F(d) \doteq x_d\right]\!\!\right] \\
&= \bigcap_{d \in D} [\![F(d) \doteq x_d]\!] \\
&= \bigcap_{d \in D} \{G \in \mathcal{F} \mid F(d) = G(d)\} \\
&= \{F\}.
\end{aligned}$$

This formula can now be used to make claims relating to a specific fact situation F. For example, for two fact situations $F, G \in \mathcal{F}$ the question of whether $F \preceq G$ corresponds to the existence of a satisfying assignment for the formula $x \doteq G \wedge \phi_s(F)$, as

$$[\![x \doteq G \wedge \phi_s(F)]\!] = [\![x \doteq G]\!] \cap [\![\phi_s(F)]\!] = \{G\} \cap \uparrow_s F.$$

In other words, we have that $x \doteq G \wedge \phi_s(F)$ is satisfiable if and only if $\{G\} \cap \uparrow_s F$ is nonempty, which is another way of saying that $F \preceq_s G$. In a similar way, we can check whether $\mathcal{C}, F \models s$ for some F by checking satisfiability of the formula $x \doteq F \wedge \Phi_s(\mathcal{C})$.

Next, we consider how to phrase whether $(F, s) \in \mathcal{C}$ is a landmark case. Let

$$\lambda_s(F) := F \doteq x \wedge \neg\Phi_s(\mathcal{C} \setminus \{(F, s)\}). \tag{4}$$

This formula states that the variable fact situation x is equal to F, and that $\mathcal{C} \setminus \{(F, s)\}$ does not force x for s. Again we can use the equations of Lemma 6 to show that this formula has the intended semantics.

Lemma 7. *A case (F, s) is a landmark iff $\lambda_s(F)$ is satisfiable.*

Proof. As in the previous results, we simply apply the equations for the semantics function:

$$\begin{aligned}
[\![\lambda_s(F)]\!] &= [\![F \doteq x \wedge \neg\Phi_s(\mathcal{C} \setminus \{(F, s)\})]\!] \\
&= [\![F \doteq x]\!] \cap [\![\neg\Phi_s(\mathcal{C} \setminus \{(F, s)\})]\!] \\
&= \{F\} \cap (\mathcal{F} \setminus [\![\Phi_s(\mathcal{C} \setminus \{(F, s)\})]\!]) \\
&= \{F\} \setminus \uparrow_s (\mathcal{C}_s \setminus \{F\}) \\
&= \begin{cases} \{F\} & \text{if } F \notin \uparrow_s (\mathcal{C}_s \setminus \{F\}), \\ \emptyset & \text{otherwise.} \end{cases}
\end{aligned}$$

□

Remark 5. Note that Corollary 1 tells us $\Phi_s(\mathcal{C})$ and $\Phi_s(\mathcal{L})$ are logically equivalent, because

$$\Phi_s(\mathcal{C}) \equiv \Phi_s(\mathcal{L}) \iff [\![\Phi_s(\mathcal{C})]\!] = [\![\Phi_s(\mathcal{L})]\!] \iff \uparrow_s \mathcal{C}_s = \uparrow_s \mathcal{L}_s.$$

This means we can freely interchange these formulas, which can be computationally advantageous if there are significantly fewer landmarks than regular cases. Of course, this does incur the overhead of computing the set of landmarks $\mathcal{L}$, which may itself be resource intensive. In the remainder of this work we may write Φ_s instead of $\Phi_s(\mathcal{C})$ or $\Phi_s(\mathcal{L})$.

Lastly, we mention that case base consistency and completeness are now easily phrased using the logical language, as the following proposition shows.

Proposition 5. *$\mathcal{C}$ consistent iff $\Phi_0 \wedge \Phi_1$ is unsatisfiable, and complete iff $\Phi_0 \vee \Phi_1$ is valid.*

Proof. We apply the semantics function of Lemma 6 and then appeal to Lemma 4:

$$\begin{aligned}
&\Phi_0 \wedge \Phi_1 \text{ is unsat} \iff \Phi_0 \wedge \Phi_1 \equiv \bot \iff [\![\Phi_0 \wedge \Phi_1]\!] = [\![\bot]\!] \iff \downarrow\mathcal{C}_0 \cap \uparrow\mathcal{C}_1 = \emptyset, \\
&\Phi_0 \vee \Phi_1 \text{ is valid} \iff \Phi_0 \vee \Phi_1 \equiv \top \iff [\![\Phi_0 \vee \Phi_1]\!] = [\![\top]\!] \iff \downarrow\mathcal{C}_0 \cup \uparrow\mathcal{C}_1 = \mathcal{F}.
\end{aligned}$$

□

2.4 A Case Base as a Binary Classifier

As the last of our theoretical considerations we investigate the relation between a case base and the concept of a classifier from machine learning; i.e. an algorithm that sorts a set of input data into one or more classes. A case base, together with the notion of forcing of Definition 2, can be considered as a classifier that can assign 0 or 1 to a new fact situation. This is also the view adopted in the work by Liu et al. (2022) and Odekerken et al. (2023). In fact, the a fortiori model has been implemented in a human-in-the-loop decision support system for web shop classification at the Dutch national police force (Odekerken & Bex, 2020). It is therefore of interest to further examine the theoretical relation between the a fortiori model, and binary classifiers in general.

Formally a binary classifier on a set A is a function $f : A \to \{0, 1\}$. The set A contains the input data, and each element $a \in A$ is assigned a label $f(a)$ which is either 0 or 1. Set-theoretically speaking, a function $f : A \to B$ with domain A and codomain B is a set of ordered pairs $\{(a, b) \in A \times B \mid f(a) = b\}$. In other words, f is a relation $f \subseteq A \times B$. However, not every relation between A and B is a function. In order for a relation $R \subseteq A \times B$ to qualify as a function it should satisfy the following criteria.

Definition 6. A relation $R \subseteq A \times B$ between sets A and B is *well-defined* if $R(a, b)$ and $R(a, b')$ implies $b = b'$, and *total* if for every $a \in A$ there is some $b \in B$ such that $R(a, b)$. When R is both well-defined and total we say it is *functional*, and write $R : A \to B$.

A relation $R \subseteq A \times B$ is functional if it associates each element in A to precisely one element of B. Given a case base $\mathcal{C}$ we define a relation $c \subseteq \mathcal{F} \times \{0, 1\}$ by $c := \{(F, s) \mid \mathcal{C}, F \vDash s\}$, so c is the forcing relation between facts and sides for a given case base $\mathcal{C}$. The question now is under what conditions c is a function $c : \mathcal{F} \to \{0, 1\}$, i.e. when is c a binary classifier? Spelling out the condition of being well-defined of Definition 6 for the relation c, we have that c is well-defined if for a fact situation F, and outcomes s and t, we have that $\mathcal{C}, F \vDash s$ and $\mathcal{C}, F \vDash t$ implies $s = t$. In other words, c is well-defined exactly when the case base is consistent. Similarly, to say that c is total is just to say that $\mathcal{C}$ is complete.

We have discussed several equivalent formulations of case base consistency and completeness, corresponding to the different views of the a fortiori models discussed in the preceding sections, and we summarize them in the following proposition.

Proposition 6. *The following are equivalent statements about consistency of a case base $\mathcal{C}$:*

(1) *$\mathcal{C}$ is consistent;*

(2) *There is no fact situation F such that $\mathcal{C}, F \vDash 0$ and $\mathcal{C}, F \vDash 1$;*

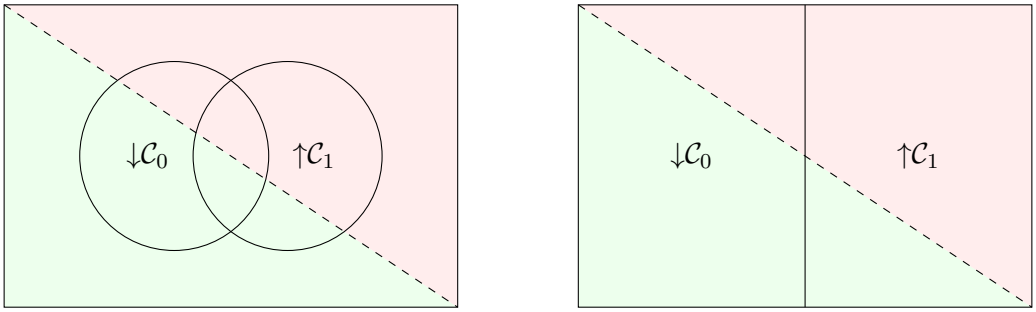
(3) *$\downarrow\mathcal{C}_0 \cap \uparrow\mathcal{C}_1 = \emptyset$;*

(4) *$\Phi_0 \wedge \Phi_1$ is unsatisfiable;*

(5) *The classify relation c is well-defined.*

Dually, we have the following list of equivalent statements expressing completeness of $\mathcal{C}$:

(1) *$\mathcal{C}$ is complete;*



(a) Neither consistent nor complete (b) Both consistent and complete

Figure 3: An adaptation of the Euler diagrams in Figure 2 for when the space of fact situations $\mathcal{F}$ is partitioned by $\mathcal{F}_0$, the green shaded area, and $\mathcal{F}_1$, the red shaded area, indicating ground truth labels 0 and 1, respectively.

(2) *For every fact situation* F *either* $\mathcal{C}, F \models 0$ *or* $\mathcal{C}, F \models 1$*;*

(3) $\downarrow\mathcal{C}_0 \cup \uparrow\mathcal{C}_1 = \mathcal{F}$*;*

(4) $\Phi_0 \vee \Phi_1$ *is valid;*

(5) *The classify relation* c *is total.*

When considering classifiers, one is often interested in classification accuracy. When the set of fact situations $\mathcal{F}$ comes with ground truth labels, it is partitioned by two sets $\mathcal{F}_0 \cup \mathcal{F}_1 = \mathcal{F}$ indicating these labels. In such a scenario, we can consider the degree to which the labels forced by the case base are in agreement with these ground truth labels. We can visualize this comparison by modifying the Euler diagram representation that we saw in Figure 2. We do this by indicating the subset $\mathcal{F}_0 \subseteq \mathcal{F}$ as a green shaded area, and the subset $\mathcal{F}_1 \subseteq \mathcal{F}$ as a red shaded area, divided by a dashed line; see Figure 3.

The Euler diagram corresponding to the general case, when the case base is neither consistent nor complete, is depicted by Figure 3a. If, however, the case base is a proper classifier (meaning it is consistent and complete), the picture looks as in Figure 3b. We can think of this Euler diagram as a confusion matrix: $\downarrow\mathcal{C}_0 \cap \mathcal{F}_0$ contains the *true negative* fact situations, $\downarrow\mathcal{C}_0 \cap \mathcal{F}_1$ the *false negative* fact situations, $\uparrow\mathcal{C}_1 \cap \mathcal{F}_1$ the *true positive* fact situations, and $\uparrow\mathcal{C}_1 \cap \mathcal{F}_0$ the *false positive* fact situations. We will return to this representation for our data analysis in Section 3.

3. Experimentally Evaluating the Model on Data

In the first half of this paper, we connected the theory of the a fortiori model developed by Horty (2011) to order theory and many-sorted logic. This second part of the paper will be about concretely applying the model to various datasets and evaluating the degree to which the model fits the data. To do this we first use the reformulation of the model in terms of many-sorted logic, described in Section 2.3.2, to write a Python implementation on the basis of the Satisfiability Modulo Theories (SMT) solver Z3 (de Moura & Bjørner, 2008). In Section 3.1, we describe how this implementation works, and then use it to fit the

Dataset	Size	Pearson Corr.		Logistic Regr.	
		$\|\mathcal{L}\|$	Cons.	$\|\mathcal{L}\|$	Cons.
Churn	7,010	1,259	59.2%	6,009	95.6%
Admission	500	41	80.2%	90	91.2%
Mushrooms	8,124	23	98.8%	23	100%
COMPAS (full)	5,873	88	8.1%	–	–
COMPAS (simp.)	1,342	12	4.2%	–	–
CORELS	907	6	100%	–	–
Tort	1,024	18	98.6%	–	–
Welfare (full)	99,988	634	71.1%	462	48.5%
Welfare (simp.)	32,876	10	67.3%	5	66%

Table 2: An overview of the various datasets used in our experiments. For each dataset we list its size, number of landmarks, and its consistency percentage. We do this for both the Pearson correlation and logistic regression methods for determining the dimension orders. In some cases both methods produce the same dimension orders, which means that all statistics will also be the same; such duplicate statistics are replaced by dashes.

model to various datasets. We evaluate various questions such as: is the dataset consistent and/or complete? If not, what is causing the inconsistency or incompleteness? If the dataset is inconsistent, how many of its cases are inconsistent? How many landmarks does the data contain, and what do they look like? We also compare different ways of automatically determining the dimension orders and the effect that they have on the aforementioned statistics. Some of the datasets we look at have known ground truth labels, which allows us to analyze exactly how well the model fits the data.

Our experiments can be roughly divided into three parts. First, in Section 3.2, we look at three datasets used in the experiments by Prakken and Ratsma (2022). We do this so we can compare the output of our implementation to known results, and so that we can test a new method for automatically learning appropriate dimension orders from the data. Secondly, in Section 3.3, we fit the model to the well-known COMPAS recidivism dataset published by Angwin et al. (2016), as well as on several variations of this dataset. This dataset consists of real-world data which is representative of the domain on which we would like to apply XAI methods based on the a fortiori model. As such, the results of this experiment are indicative of the feasibility of such XAI methods. Thirdly, in Section 3.4, we consider datasets used by Steging et al. (2021). We use these because they have known ground truth labels, which allows us to precisely evaluate the model's fit to the data. An overview of our findings is given in Table 2.

3.1 A Software Implementation of the a Fortiori Model

In this section, we describe how we can implement the a fortiori model in Python using the SMT solver Z3 (de Moura & Bjørner, 2008). In order to be able to compute with the a fortiori model we require two main components. First—to construct the model—we need a

method for determining the dimension orders. Secondly, we need a way to operationalize it, so that we can compute, for example, whether some new fact situation has its outcome forced by a case base. Other necessary ingredients like data representations can be handled with built-in Python functionality. We start by describing how we determine the dimension orders in Section 3.1.1, and then how we use Z3 to operationalize the model in Section 3.1.2.

3.1.1 Determining Dimension Orders

Determining appropriate orders for the dimensions is not a straightforward task. They constitute an assumption that the values along the dimension tend to prefer either of the binary outcomes. For instance, in our example with recidivism data we have an `Age` dimension, and to determine its order is to say whether the elderly are more likely to recidivate than the young, or vice versa. Knowledge engineering techniques and statistical methods can be used for this purpose. For instance, for the `Age` dimension, much has been written on the interplay between age and recidivism, the conclusion of which is summarized by the adage that "people age out of crime", meaning that as people age they become decreasingly likely to recidivate. Another option is to look at statistical trends in the data, for instance, by considering the sign of the Pearson correlation between age and recidivism. If it is positive, we can say that likelihood of recidivism increases with age, and if it is negative, we can say it decreases.

For our implementation, we employ the statistical method. We will use the same underlying idea as used by Prakken and Ratsma (2022), which is to use a function c that associates each numerical feature x with a *coefficient* $c(x)$ indicating the degree to which the values of x favor outcome 1. If $c(x)$ is positive we order the values of x with the usual 'less-than' order $\leq$ on the number line, and if it is negative we order it using the 'greater-than' order $\geq$; so more precisely $\preceq := \leq$ if $c(x) \geq 0$ and $\preceq := \geq$ if $c(x) < 0$.

If x is categorical we cannot apply c directly so we use *dummy variables*. More specifically, if x is a categorical feature that can take the possible (unordered) values $v_1, \ldots, v_n$, then we introduce for each value v_i a dummy variable d_{v_i} which is a binary feature indicating whether $x = v_i$. Then we define $v_i \preceq v_j$ if and only if $c(d_{v_i}) \leq c(d_{v_j})$.

Prakken and Ratsma (2022) define c on the basis of Pearson correlation, but for the present work we define c using logistic regression. Supposing we have features $x_1, \ldots, x_n$ the logistic model has parameters $\beta_0, \ldots, \beta_n$, and models the probability that a given sample belongs to class 1 by the formula

$$p(x_1, \ldots, x_n) = \frac{1}{1 + e^{-(\beta_0 + \sum_i \beta_i x_i)}}. \tag{5}$$

We find appropriate values for the β parameters using the scikit-learn implementation of a maximum likelihood estimation with default parameters (Pedregosa et al., 2011), and after this is done we can simply put $c(x_i) := \beta_i$.

As mentioned we opt to use logistic regression rather than Pearson correlation. There are several reasons for this. Firstly, logistic regression seems to be a better choice conceptually, since it optimizes the coefficients in tandem rather than compute them independently of one another. Secondly, logistic regression seems to perform better in practice, as we will demonstrate in the coming sections. Lastly, the method using Pearson correlation seems to work poorly with categorical features, as we will now illustrate.

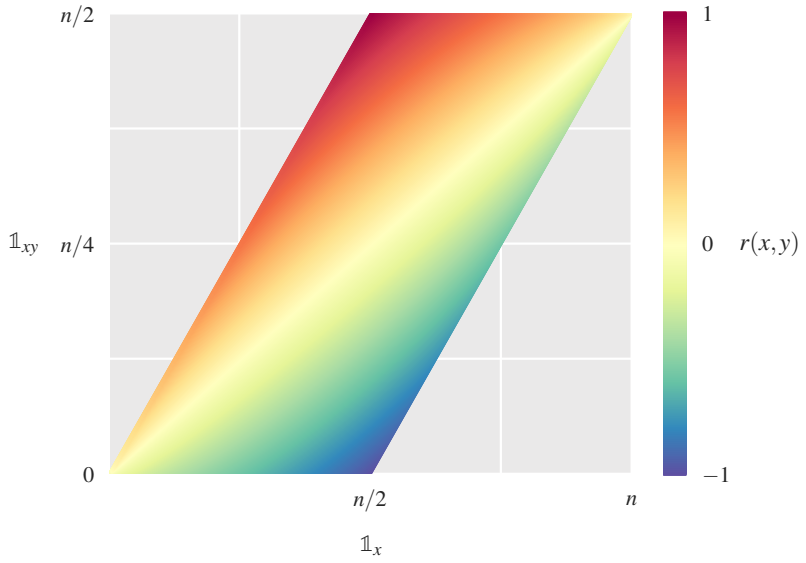


Figure 4: A plot of Eq. (6), the Pearson correlation coefficient for binary vectors x and y for a fixed value of $n := 400$ and with $\mathbb{1}_y := n/2$. The gray area marks points that violate one of the inequalities $\mathbb{1}_x + \mathbb{1}_y - n \leq \mathbb{1}_{xy} \leq \mathbb{1}_x$, and as such could not result from a sample.

Given n samples $(x_1, y_1), \ldots, (x_n, y_n)$ of binary variables x and y, the estimate of the Pearson correlation $r(x, y)$ between x and y is given by

$$r(x,y) = \frac{n\mathbb{1}_{xy} - \mathbb{1}_x\mathbb{1}_y}{\sqrt{n\mathbb{1}_x - \mathbb{1}_x^2}\sqrt{n\mathbb{1}_y - \mathbb{1}_y^2}}, \tag{6}$$

where $\mathbb{1}_x$ is the number of times x takes value 1 in the samples, $\mathbb{1}_y$ the number of times y takes value 1, and $\mathbb{1}_{xy}$ the number of times x and y both take value 1.

In order to get a sense of how this function behaves we plot its values for a fixed n and with $\mathbb{1}_y := n/2$, see Figure 4. This plot shows that when $\mathbb{1}_x$ is relatively low, or relatively high, the range of $r(x, y)$ (as a function of $\mathbb{1}_{xy}$) is not $[-1, 1]$ but some restricted interval near 0. More precisely, writing $s(x) := \sqrt{n\mathbb{1}_x - \mathbb{1}_x^2}$, we can calculate that for $0 \leq \mathbb{1}_x \leq n/2$ the range of $r(x, y)$ is $[-\mathbb{1}_x/s(x), \mathbb{1}_x/s(x)] \approx [-\mathbb{1}_x/(n/2), \mathbb{1}_x/(n/2)]$, i.e. is roughly proportional to $\mathbb{1}_x$. This is undesirable when x is a dummy variable, as then $\mathbb{1}_x$ simply indicates the number of times the original categorical feature took the value which the dummy variable represents, i.e. the number of samples we have of that class.

Example 6. Let us consider an example to illustrate this point. The original COMPAS data includes a `Race` variable, with possible values including 'Asian' and 'Caucasian'. The value Asian occurs much less often than Caucasian (0.4% against 34%), meaning that the value of $\mathbb{1}_x$ for the dummy variable for `Race` = Asian is much lower than that for the `Race` = Caucasian variable. As a result, its Pearson correlation must land in a very small interval around 0, while the one for Caucasian has almost the full range available. Indeed, the order for the `Race` dimension on the basis of the Pearson correlation method puts Caucasian in the last position (i.e. comparatively least prone to recidivate), and Asian a little over halfway in the order. To compare this with a measure that does not place such

great importance on the number of samples that we have of each race, we consider the relative frequency $\mathbb{1}_{xy}/\mathbb{1}_x$ of recidivism within that class. The picture is now the opposite of what we see with Pearson correlation, with Asian ending lowest in the ranking (at 28% prevalence) and Caucasian a little over halfway (at 40% prevalence).

3.1.2 Using Z3 to Operationalize the Model

Satisfiability Modulo Theories (SMT) is about procedurally checking the satisfiability of formulas over a theory, in the sense described in Section 2.3. In particular: given a many sorted signature Σ, a formula $\phi \in L^\Sigma$, a structure $\mathbb{A}$ and a theory $T \subseteq L^\Sigma$, an SMT solver is concerned with deciding whether there is a satisfying assignment $\alpha \in [\![\phi]\!]$ or not. Using the equivalence between validity of ϕ and unsatisfiability of $\neg\phi$ this means an SMT solver can also be used to check validity. The decidability and complexity of answering this satisfiability problem greatly depend on the theories in question. Bradley and Manna (2007, Section 3) have given an overview of some supported theories and their complexities.

In this section, we describe how we use Z3—a state-of-the-art SMT solver—to answer questions for a given set D of dimensions and a case base $\mathcal{C}$, such as:

(1) Given two fact situations $F, G \in \mathcal{F}$ does $F \preceq_s G$ holds?

(2) For a fact situation F and an outcome s, does $\mathcal{C}, F \vDash s$ hold?

(3) Is $\mathcal{C}$ consistent and/or complete?

(4) Given a case $(F, s) \in \mathcal{C}$, is (F, s) a landmark of $\mathcal{C}$?

Answering these questions with Z3 is a relatively straightforward application of our work in Section 2.3.2. For instance, to answer question (1) we use the formula $G \doteq x \wedge \phi_s(F)$ of Eqs. (1) and (3) because Z3 can determine whether there is a satisfying assignment in $[\![G \doteq x \wedge \phi_s(F)]\!] = \{G\} \cap \uparrow_s F$, which is inhabited if and only if $F \preceq_s G$. Similarly, to see if F is forced by $\mathcal{C}$ for s we use the formula $F \doteq x \wedge \Phi_s$, since $[\![F \doteq x \wedge \Phi_s]\!] = \{F\} \cap \uparrow_s \mathcal{C}$ is inhabited if and only if $\mathcal{C}, F \vDash s$. To answer question (3) we can use Proposition 5; the case base is inconsistent if and only if $\Phi_0 \wedge \Phi_1$ is satisfiable, and incomplete if and only if $\neg(\Phi_0 \vee \Phi_1)$ is satisfiable. Finally, to check whether a case (F, s) is a landmark of $\mathcal{C}$ we can use the formula $\lambda_s(F)$ of Eq. (4), since $[\![\lambda_s(F)]\!]$ is inhabited if and only if $(F, s) \in \mathcal{L}$.

The reader familiar with Python can find a simple example of how our implementation with Z3 works in Appendix A. The full implementation is available online.[2]

3.2 The Churn, Admission, and Mushroom Datasets

We begin by repeating the experiment by Prakken and Ratsma (2022, Section 6) on the Churn,[3] Mushroom (Schlimmer, 1981), and Admission datasets (Acharya et al., 2019). All three of these datasets are, or at least appear to be, largely synthetic. The Churn dataset contains "information about a fictional telco company that provided home phone and Internet services to 7,043 customers in California in Q3." The Mushroom dataset contains

2. https://github.com/wijnanduu/AFCBR.
3. https://community.ibm.com/community/user/businessanalytics/blogs/steven-macko/2019/07/11/telco-customer-churn-1113.

"descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family." Lastly, the Admission dataset contains information about the chance of university admission on the basis of data like undergraduate GPA. Again, this dataset seems to contain at least some synthetic elements, as its author writes that it had values "entered manually with no specific pattern. It was random assignment." A more extensive description of these datasets and their features can be found in the work by Prakken and Ratsma (2022).

We report the findings of our implementation in Table 2, which can be compared to the results found by Prakken and Ratsma (2022, Table 3). We list the number of landmarks $|\mathcal{L}|$ as well as the consistency percentage, which is computed as the relative frequency of consistent cases in the dataset:

$$\text{Cons}(\mathcal{C}) := 100 \cdot \left(1 - \frac{|\mathcal{C}_0 \cap \uparrow\mathcal{C}_1| + |\mathcal{C}_1 \cap \downarrow\mathcal{C}_0|}{|\mathcal{C}|}\right).$$

We find an identical consistency percentage for the Mushrooms dataset, but only approximately equal percentages for the Churn and Admission datasets. The difference in the percentage for Churn is because Prakken & Ratsma did not delete duplicate occurrences of cases. We did delete duplicate cases for the sake of our landmark analysis; if two cases have identical fact situations and outcomes, but are not considered equal, then they will 'force' each other's outcome and so are not considered landmarks when they otherwise might have been. The consistency percentage on the Admission dataset also differs, even though the number of cases there is equal. It is not entirely clear why this is. Since the difference in percentages is small—only 0.4%—and the results are otherwise in agreement, we do not further investigate the source of this difference.

As we can see, the approach using logistic regression tends to increase both the number of landmarks as well as the consistency percentage—in the case of the Churn dataset by as much as 36.4%. This suggests to us that logistic regression is indeed a better method for the purpose of automatically assigning dimension orders to the features.

3.3 The COMPAS Recidivism Dataset

For our second evaluation of the model we use the COMPAS recidivism dataset, published by Angwin et al. (2016), which contains information on convicts and whether they recidivated within two years after being arrested for an initial charge. We chose this dataset because it consists of real-world data that is closely related to the type of situations for which we want to develop XAI methods: data-driven methods with legal, ethical, or social impact to end users.

For this evaluation we proceed just as we did for the Churn, Mushroom, and Admission datasets—we fit a logistic regression model to the data to determine the dimension orders and subsequently evaluate various statistics to measure the degree to which the a fortiori model fits the data. However, unlike for the aforementioned datasets, we need to do more extensive preprocessing in order to get the data in an appropriate format. This results in a dataset that we will refer to as the 'COMPAS dataset'. In order to get a better understanding of our experimental results, we also make two variations on this dataset. The first, which we call the 'simplified COMPAS dataset', contains only a subset of the features of the COMPAS set. Then, we relabel the simplified version according to a rule found by Angelino et al.

Table 3: An overview of the COMPAS features of interest. Angwin et al. (2016) did not give a comprehensive overview of the meaning of all the features used in their analysis, so we should note that this is only our best attempt at an interpretation.

Feature	Description	Order
`Age`	Age of the convict at the time of the COMPAS assessment.	Descending
`Sex`	Gender as specified when the convict was arrested, can take on the values 'Male' or 'Female'.	Female $\prec$ Male
`ChargeDegree`	Indicates whether the charge that led to the assessment was a felony (F) or a misdemeanor (M).	M $\prec$ F
`DaysInJail`	The number of days the convict spends in jail for the crime, computed by comparing (and rounding down) the number of days between the `c_jail_in` and `c_jail_out` fields.	Ascending
`DaysInCustody`	The number of days the convict spends in custody, computed in the same way as `DaysInJail` but with the `c_custody_in` and `c_custody_out` fields.	Ascending
`Priors`	The number of offenses committed prior to the one that led to the COMPAS assessment. The value of this field is computed as the sum of the values of `juv_fel_count`, `juv_misd_count`, `juv_other_count`, and `priors_count` fields in the original dataset.	Ascending
`Label`	The label, indicating whether there was "a criminal offense that resulted in a jail booking and took place after the crime for which the person was COMPAS scored [...] within two years after the first." (Larson et al., 2016)	N/A

(2018) using their Certifiably Optimal Rule Lists (CORELS) algorithm. We name this last dataset the 'CORELS dataset'.

The preprocessing steps we took are described in Section 3.3.1. We then describe in Section 3.3.2 our results for the COMPAS dataset, in Section 3.3.3 our results for the simplified COMPAS dataset, and in Section 3.3.4 our results for the CORELS dataset.

3.3.1 Data Preprocessing

Before analyzing the COMPAS data we preprocess it. In particular, we discard features that are not of interest, delete rows that do not have values for the remaining features, create new features on the basis of old ones, and finally delete duplicate rows. Below follows a more detailed description of the steps taken.

if $(age = 18 - 20)$ **then predict** *yes*
else if $(age = 21 - 23)$ **and** $(priors = 2 - 3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

Figure 5: A rule list for the COMPAS dataset found by Angelino et al. (2018) using their CORELS algorithm. The clause related to sex has been excluded since this feature is omitted from the simplified COMPAS dataset for the sake of visualizability.

Table 4: On the left is a summary of the strength order on the COMPAS dataset, and the impact of the landmarks l_0 and l_1 defined in Definition 7. On the right is a concrete description of l_0 and l_1. Notice that they are archetypal examples of the *opposite* class that they belong to; l_0 is a young male with many priors, who did not recidivate; while l_1 is an older female with no priors, who did recidivate.

Property	**Label 0**	**Label 1**	**Total**
Consistent	76	397	473
Inconsistent	2,783	2,617	5,400
Forced by l_0	2,271	1,765	4,036
Forced by l_1	2,296	2,700	4,969
Landmark	70	18	88

d	$l_0(d)$	$l_1(d)$
`Age`	23	49
`Sex`	Male	Female
`ChargeDegree`	F	M
`DaysInJail`	70	0
`DaysInCustody`	70	0
`Priors`	11	0

First, we discard features that are not of interest. For instance, many of the features in the original dataset pertain to the COMPAS system, but presently we are only interested in the data describing the convicts and whether they recidivated or not, not in the COMPAS system itself. For example, one of the features describes the recidivism risk score (on a 1–10 scale) which COMPAS assigned to the individual.

Some features are of interest to us but are not in the right format. For instance, the two columns `c_jail_in` and `c_jail_out` together tell us how many days the convict spend in prison, but are represented in a date format, so we replace them with a new `DaysInJail` feature holding the number of days spent in prison. A complete overview of the resulting features and their meaning can be found in Table 3.

Lastly, we remove any rows that do not have values for any of the relevant features, or which occur more than once in the data. This last step is necessary for our landmark analysis; a case c may be a landmark, but if there is a second case d with exactly the same fact situation and outcome as c but not *equal* to c, then neither c nor d are landmarks.

We are then left with a total of 5,873 rows and we will henceforth refer to that set when we say 'COMPAS dataset'. In addition, we will look at two variations on that set. The first we will call the 'simplified COMPAS dataset', which is obtained from the COMPAS dataset by omitting all features except `Age` and `Priors`, and then deleting all duplicates. The second we call the 'CORELS dataset', and is obtained by changing the labels in the simplified COMPAS dataset according to the recidivism prediction rule found by Angelino et al. (2018, Figure 1) using their CORELS algorithm, see Figure 5.

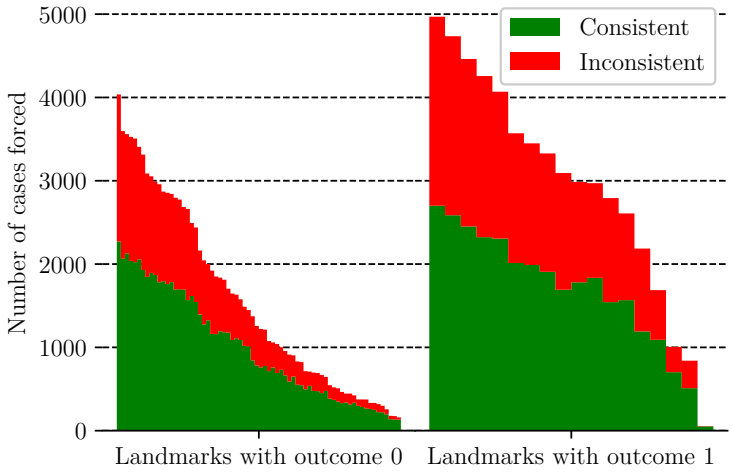


Figure 6: A visualization of the impact of the landmarks in the COMPAS data. Each vertical bar represents one landmark and shows the number of cases for which it forces the decision. The green area indicates the cases with an outcome equal to that of the landmark, and the red area the cases with an outcome different from the landmark (and which are therefore made inconsistent by the landmark). More precisely, for each landmark $(F, s) \in \mathcal{L}$ the green area represents $|\mathcal{C}_s \cap \uparrow_s F|$ and the red area represents $|\mathcal{C}_{\bar{s}} \cap \uparrow_s F|$.

3.3.2 Results on the COMPAS Dataset

Having selected the dimensions, assigned their orders, and constructed the case base, we can now evaluate various statistics. We start by looking at the consistency percentage, i.e. the relative frequency of cases that do not have their outcome disputed by the strength order on the case base. We find the COMPAS dataset is only 8% consistent, see Table 2. This low percentage is caused by a small number of landmarks—outliers in the data that one would expect to have the opposite label of the one they received. We identify two landmarks l_0 and l_1 as being most impactful, which are defined as follows.

Definition 7. Given a finite case base $\mathcal{C}$ and an outcome s we define the set L_s of cases with outcome s that force the outcome of the greatest number of other cases in $\mathcal{C}$:

$$L_s := \operatorname{argmax}_{F \in \mathcal{C}_s} |\uparrow_s F \cap (\mathcal{C}_0 \cup \mathcal{C}_1)|.$$

When L_s is a singleton we write l_s for its sole element.

By transitivity of the strength order the cases in L_s are also landmarks, i.e. we have $L_s \subseteq \mathcal{L}_s$. In the datasets we consider in this work the L_s sets are singletons, so we will just refer to their sole elements l_0 and l_1. The l_s cases in the COMPAS dataset are shown in Table 4. In Figure 6 an overview of the collective impact of the landmarks is shown.

Remark 6. The notions of landmark and outlier, while similar, are not quite the same: a landmark need not be an outlier (cf. Figure 8) and an outlier need not be a landmark (for instance, when there is an outlier even further across the best-fit decision boundary).

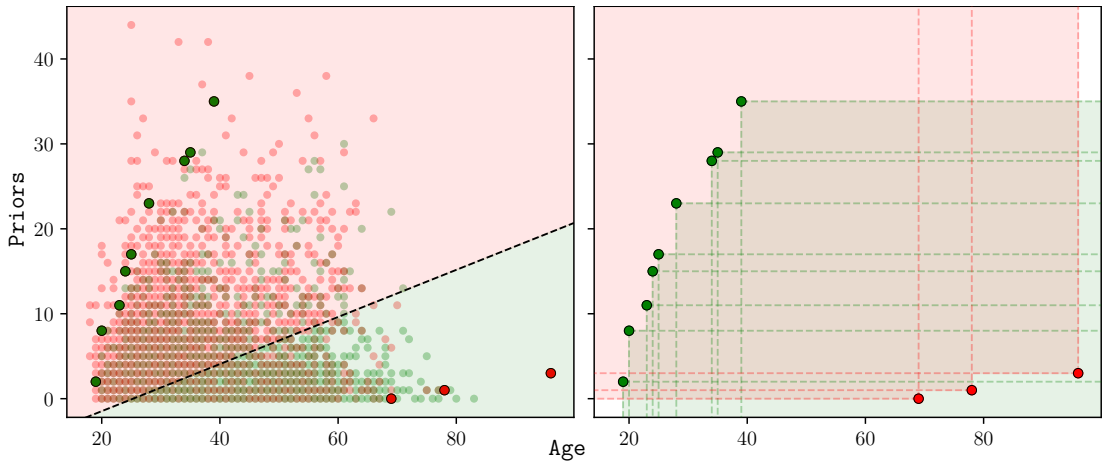


Figure 7: Two illustrations of the simplified COMPAS dataset. The green dots correspond to cases with outcome 0, and the red dots to those with outcome 1. The enlarged circles indicate the landmarks. On the left, all cases in the case base are shown, together with a dotted line indicating the decision boundary associated with the logistic regression coefficients. On the right, only the landmarks are shown, together with their forcing cones.

3.3.3 Results on the Simplified COMPAS Dataset

High dimensional data is difficult to visualize, so in order to get a better view of these results we repeat our analysis on a subset of the data with only the two most predictive variables—`Age` and `Priors`. We call this the simplified COMPAS dataset. The resulting order on the variables remains the same as in the larger version. This lets us visualize the data, the decision surface of our logistic model, and the landmarks; see Figure 7 for the resulting plot. The landmarks highlight the cause for the inconsistency: there are many cases that lie on the opposite side of the decision boundary for their class, causing large overlap in their forcing cones.

3.3.4 Results on the CORELS Dataset

The preceding results have shown that the model of precedential constraint is a poor fit on the COMPAS data. This makes sense intuitively, because when someone of a certain age and with some number of priors recidivates, we cannot expect this to set a precedent that future convicts will abide by. For example, when an elderly lady with no prior offenses recidivates, this will have very little influence on the behavior of convicts thereafter. In other words, the process underlying recidivism does not respect precedence.

This type of reasoning should be more suited to our running example from Section 2 in which we judge *risk* of recidivism. When a person is assigned low or high risk of recidivism, we would expect this assignment to obey the a fortiori principle.

To test this hypothesis we change the labels of the simplified COMPAS data according to a sensible risk assessment rule, mined from the original COMPAS data by Angelino et al. (2018, Figure 1), as a demonstration of their CORELS algorithm. This rule is listed in Figure 5, with the only modification being that we omit the clause related to sex from the first case

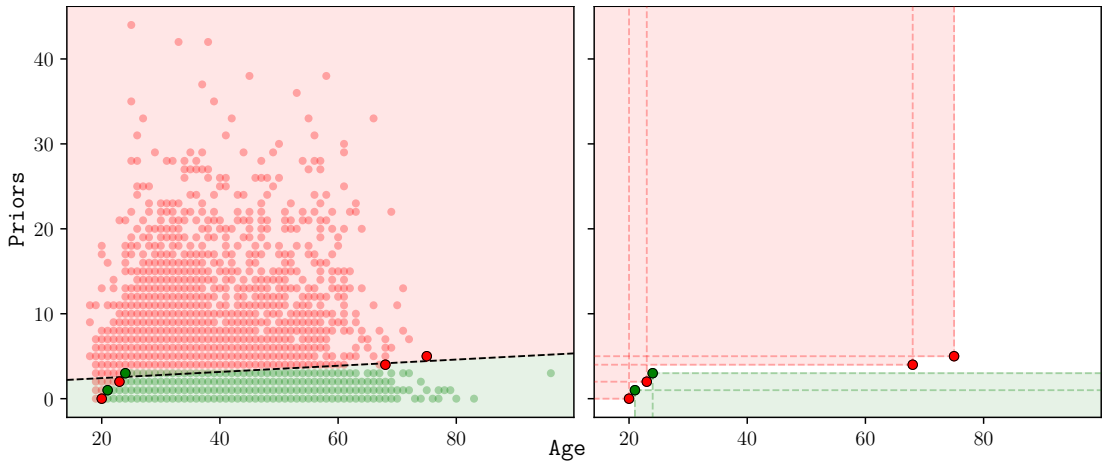


Figure 8: Two illustrations of the simplified COMPAS dataset. The green dots correspond to cases with outcome 0, and the red dots to those with outcome 1. The enlarged circles indicate the landmarks. On the left, all cases in the case base are shown, together with a dotted line indicating the decision boundary associated with the logistic regression coefficients. On the right, only the landmarks are shown, together with their forcing cones.

distinction since we have omitted this feature for the sake of visualizability. Changing all labels according to this rule, and then removing duplicates, results in a new dataset that we refer to as the COMPAS dataset.

Now we again fit our model to this data and visualize the decision boundary of the logistic regression model, along with the forcing cones of the landmarks; see Figure 8 for the resulting plot. As expected, the decision rule of Figure 5 does satisfy the a fortiori principle, and as a result the consistency is very high (in fact the dataset is fully consistent). The forcing cones of the landmarks are in agreement with the decision boundary determined by the logistic regression analysis.

In all, our results on the COMPAS datasets suggest that we can think of the phenomenon of inconsistency in two ways. The first is the mathematical view that the theory of precedential constraint contains a linearity assumption and that the consistency percentage is a measure of the degree to which the data is linearly separable. Of each class, the landmarks are then those cases that lie furthest in the direction of the best fit linear decision boundary, and the farther they cross it the more inconsistency they cause. The second is the semantic view that tells us to what degree the labelling process relies on a fortiori reasoning, or the degree to which we can expect precedent to be obeyed. If this is the case, then the landmarks are those cases that most reveal the nature of the underlying labelling process.

Our results also suggest that the presence of a small number of landmarks that force the decision of the rest is what we can expect of an average dataset, because in general a partial order will have far fewer minimal elements than that it will have elements in total. Two factors that can influence this is the number of dimensions and the way in which we order them. For instance, if we have a dimension with more than two values and we order them so that they are all incomparable, it will immediately become impossible for any case to force the outcome of another, and so every case becomes a landmark.

Table 5: On the left: the landmarks in the consistent and incomplete CORELS dataset; and on the right: the landmarks of the modified, consistent and complete CORELS dataset.

Age	Priors	**Label**
21	1	0
24	3	0
20	0	1
23	2	1
68	4	1
75	5	1

$\texttt{Age}_{\leq 100}$	Priors	**Label**
21	1	0
24	3	0
20	0	1
23	2	1
100+	4	1

3.3.5 A Logical Analysis of the CORELS Dataset

An interesting fact of the CORELS dataset is that its labels are determined by a logical rule, which can be expressed in the same many-sorted language we used in Section 2.3.2 to formulate the a fortiori model. More specifically, the rule in Figure 5 corresponds to a formula $\Psi \in L^{\Sigma(D)}$ defined by:

$$\begin{aligned} \Psi &:= C_1 \vee C_2 \vee C_3, \\ C_1 &:= 18 \leq x_{\texttt{Age}} \leq 20, \\ C_2 &:= (21 \leq x_{\texttt{Age}} \leq 23) \wedge (2 \leq x_{\texttt{Priors}} \leq 3), \\ C_3 &:= 3 < x_{\texttt{Priors}}. \end{aligned} \tag{7}$$

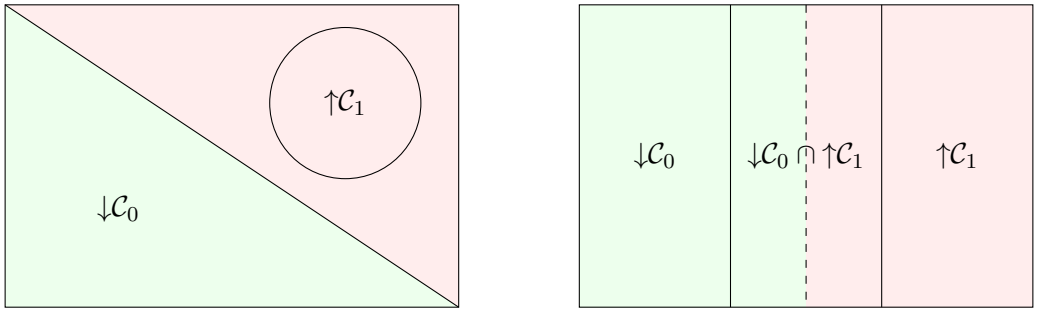
A fact situation F is assigned label 1 if $\mathbb{D}, F \vDash \Psi$, and 0 otherwise—i.e. when $\mathbb{D}, F \vDash \neg\Psi$. Letting $\mathcal{F}_1 = [\![\Psi]\!]$ and $\mathcal{F}_0 = [\![\neg\Psi]\!] = \mathcal{F} \setminus [\![\Psi]\!]$. This means we have a situation as described in Section 2.4, in which the set of fact situations $\mathcal{F}$ is equal to a disjoint union $\mathcal{F}_0 \cup \mathcal{F}_1 = \mathcal{F}$ indicating binary ground truth labels. Moreover, since these $\mathcal{F}_0$ and $\mathcal{F}_1$ sets are defined in the logical language of the a fortiori model, we can use Z3 to reason about the relation between the ground truth labels and the forcing relation induced by the CORELS case base.

Let us illustrate how this works by looking at the Φ_s formulas. The CORELS dataset contains very few landmarks—only 6 in total—which allows us to write them down; see Table 5 for an overview. This also means we can write out the corresponding Φ_s formulas:

$$\Phi_0 = (24 \leq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \leq 3) \vee (21 \leq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \leq 1), \tag{8}$$

$$\begin{aligned} \Phi_1 = (75 \geq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \geq 5) \vee (68 \geq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \geq 4) \vee \\ (20 \geq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \geq 0) \vee (23 \geq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \geq 2). \end{aligned} \tag{9}$$

Using these we can precisely analyze the degree to which the forcing relation on cases is in accordance to the ground truth labels. For instance, is it always the case that when the case base forces a fact situation F for outcome 0, that F has ground truth label 0? In other words, does the inclusion $\downarrow\mathcal{C}_0 \subseteq \mathcal{F}_0$ hold? And what about the converse, $\downarrow\mathcal{C}_0 \supseteq \mathcal{F}_0$? Recall from Section 2.3.2 that these questions have logical counterparts. To check $\downarrow\mathcal{C}_0 = \mathcal{F}_0$ is the same as to check that $\Phi_0 \leftrightarrow \neg\Psi$ is valid, and the CORELS dataset is simple enough that this



(a) CORELS (b) Tort law

Figure 9: Euler diagram representations of the relation between the CORELS case base and the ground truth labels determined by the decision rule of Figure 5 (in 9a), and of the relation between the tort law case base and the labels determined by Eq. (10) (in 9b).

can be done by hand, using the basic rules for manipulating logical formulas:

$$\begin{aligned}
\neg\Psi &\leftrightarrow \neg\bigvee\{18 \leq x_{\texttt{Age}} \leq 20,\\
&\qquad (21 \leq x_{\texttt{Age}} \leq 23) \wedge (2 \leq x_{\texttt{Priors}} \leq 3),\\
&\qquad 3 < x_{\texttt{Priors}}\}\\
&\leftrightarrow \bigwedge\{21 \leq x_{\texttt{Age}},\\
&\qquad (x_{\texttt{Age}} \leq 20 \vee 24 \leq x_{\texttt{Age}}) \vee (x_{\texttt{Priors}} \leq 1 \vee 4 \leq x_{\texttt{Priors}}),\\
&\qquad x_{\texttt{Priors}} \leq 3\}\\
&\leftrightarrow \bigwedge\{21 \leq x_{\texttt{Age}},\\
&\qquad 24 \leq x_{\texttt{Age}} \vee x_{\texttt{Priors}} \leq 1,\\
&\qquad x_{\texttt{Priors}} \leq 3\}\\
&\leftrightarrow (24 \leq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \leq 3) \vee (21 \leq x_{\texttt{Age}} \wedge x_{\texttt{Priors}} \leq 1)\\
&\leftrightarrow \Phi_0.
\end{aligned}$$

In other words, we apply De Morgan's law to $\neg\Psi$, simplify the resulting expressions, distribute the conjunction over the disjunction, and then finally simplify the expression again to obtain Φ_0. Thankfully, we do not have to do this by hand, as Z3 can quickly perform such verifications. Any subsequent claims that we make about the validity of formulas was checked using Z3.

Similarly to the derivation above, we can show that $\Phi_1 \rightarrow \Psi$ is valid, which tells us that $\uparrow\mathcal{C}_1 \subseteq \mathcal{F}_1$. However, as we saw in Proposition 2, this inclusion is necessarily strict. An Euler diagram representation for the CORELS case base can be found in Figure 9a.

The proof of Proposition 2 shows that the problem with making the CORELS case base complete is that the values for `Priors` and `Age` can become infinitely large. Since case bases are finite by definition, we can always find fact situations on the northeast part of the (`Priors`, `Age`) plane that do not have their outcome forced. If we put a cap on either of these values it would be possible to make the case base complete. Let $\texttt{Age}_{\leq 100}$ denote the dimension equal to the `Age` dimension with the exception that it has a highest value '100+', i.e. fact situations have their age represented along this dimension, and any value that would

normally be above 100 gets assigned the 100+ value. More specifically, let $\texttt{Age}_{\leq 100}$ be a dimension consisting of the set $\{18, 19, 20, \ldots, 99, 100+\}$ ordered by $\geq$. Now, the CORELS case base can be made into a complete (and consistent) case base for the $\texttt{Age}_{\leq 100}$ and `Priors` dimensions by the addition of a case with $\texttt{Age}_{\leq 100}$ value 100+ and `Priors` value 4. See Table 5 for the landmarks of the resulting case base.

3.4 The Tort and Welfare Datasets

In Section 3.3.5 we saw an example of a dataset that has its labels determined on the basis of a logical formula Ψ. The set of fact situations $\mathcal{F}$ was partitioned in two parts $\mathcal{F}_1 = [\![\Psi]\!]$ and $\mathcal{F}_0 = [\![\neg\Psi]\!] = \mathcal{F} \setminus [\![\Psi]\!]$, indicating the ground truth labels of the fact situations. This allowed us to precisely measure the fit of the a fortiori model by looking at the relationships between the sets $\uparrow_s \mathcal{C}_s$ and $\mathcal{F}_s$. In this section, we look at more datasets that come with such a labelling formula Ψ. In particular, we consider the fictional welfare benefit domain first introduced by Bench-Capon (1993), and several variations on this dataset. In addition, we look at data on a real legal setting, namely the domain of Dutch tort law (Verheij, 2017).

These datasets were recently used by Steging et al. (2021) to see if modern machine learning systems can learn the rules used to label the examples in these datasets. In this section we will essentially do the same but for the a fortiori model, through an analysis similar to the one we performed for the CORELS dataset in Section 3.3.5. We perform this analysis first for the tort dataset in Section 3.4.1, and then for the welfare datasets in Section 3.4.2.

3.4.1 The Tort Dataset

We begin by considering the dataset for the Dutch tort law domain. This is law describing when a wrongful act is committed, and when the resulting damages must be repaired. The label of the training examples is whether such a 'duty to repair' holds according to the law in that particular fact situation. Fact situations are described along 12 binary features. Examples of these features are `vun`, which states that the act was a violation of unwritten law against proper social conduct, or `imp`, which states the act can be imputed to the person that committed the act. For a complete overview of the features and their meaning the reader is referred to the work by Verheij (2017, Table 1).

This duty to repair can be formalized according to the following rule:

$$\begin{aligned}
\Psi &:= \textstyle\bigwedge_{1 \leq i \leq 5} C_i, \\
C_1 &:= x_{\texttt{cau}}, \\
C_2 &:= x_{\texttt{ico}} \vee x_{\texttt{ila}} \vee x_{\texttt{ift}}, \\
C_3 &:= x_{\texttt{vun}} \vee (x_{\texttt{vst}} \wedge \neg x_{\texttt{jus}}) \vee (x_{\texttt{vrt}} \wedge \neg x_{\texttt{jus}}), \\
C_4 &:= x_{\texttt{dmg}}, \\
C_5 &:= \neg(x_{\texttt{vst}} \wedge \neg x_{\texttt{prp}}).
\end{aligned} \tag{10}$$

The consistency percentage and number of landmarks for this dataset can be found in Table 2. Since there are only 10 binary features there are only $2^{10} = 1{,}024$ possible fact situations for this domain. The dataset we use contains all 1,024 of them, and so this case base is necessarily complete. Using Z3, we can furthermore prove that $\neg\Psi \rightarrow \Phi_0$ and $\Psi \rightarrow \Phi_1$

Table 6: A description of the features appearing in the welfare set together with a description of their meaning (Bench-Capon, 1993).

Feature	Values	Description
`Age`	0 – 100	The person's age; should be of pensionable age to be eligible (60 for a woman, 65 for a man).
`Sex`	Male or female	The person's sex, used to determine pension age.
$\texttt{Con}_1, \ldots, \texttt{Con}_5$	0 or 1	The person should have paid contributions in four out of the last five relevant contribution years.
`Spouse`	True or false	The person should be a spouse of the patient.
`Absent`	True or false	The person should not be absent from the UK.
`Resources`	0 – 10,000	The person should have capital resources not amounting to more than 3,000£.
`Type`	In or out	If the relative is an in-patient the hospital should be within a certain distance: if an out-patient, beyond that distance.
`Distance`	0 – 100	Distance to the hospital.

are valid, which means that $\mathcal{F}_0 \subseteq \downarrow\mathcal{C}_0$ and $\mathcal{F}_1 \subseteq \uparrow\mathcal{C}_1$. The corresponding Euler diagram representation can be found in Figure 9b.

3.4.2 The Welfare Datasets

Next, we turn to the welfare datasets, first used by Bench-Capon (1993) to investigate whether neural networks can handle open texture in law. They contain data about a fictional welfare benefit paid to pensioners to defray expenses for visiting a spouse in a hospital. An overview of the features appearing in this dataset can be found in Table 6. The labels are determined as a logical function Ψ of these features, defined by:

$$
\begin{aligned}
\Psi &:= \textstyle\bigwedge_{1\leq i\leq 6} C_i, \qquad (11)\\
C_1 &:= (x_{\texttt{Sex}} = \mathrm{F} \wedge x_{\texttt{Age}} \geq 60) \vee (x_{\texttt{Sex}} = \mathrm{M} \wedge x_{\texttt{Age}} \geq 65),\\
C_2 &:= 4 \leq \textstyle\sum_{1\leq i\leq 5} x_{\texttt{Con}_i},\\
C_3 &:= x_{\texttt{Spouse}},\\
C_4 &:= \neg x_{\texttt{Absent}},\\
C_5 &:= x_{\texttt{Resources}} \leq 3{,}000,\\
C_6 &:= (x_{\texttt{Type}} = \mathrm{in} \wedge x_{\texttt{Distance}} < 50) \vee (x_{\texttt{Type}} = \mathrm{out} \wedge x_{\texttt{Distance}} \geq 50).
\end{aligned}
$$

Steging et al. (2021) used several different versions of the original welfare dataset for their experiments. Amongst these are two datasets each containing 50,000 examples, randomly sampled in the ranges described in Table 6, and each labelled according to the formula Ψ in Eq. (11). These were designed to either fail on a random number of the conditions

$C_1, \ldots, C_6$, or to fail on just one specific condition. For our purposes, this distinction is not important, so we merge the datasets into one set that we will henceforth refer to as the welfare dataset. After merging and removing duplicates it contains 99,988 cases; its number of landmarks and consistency percentage can be found in Table 2.

Interesting to note is that the Pearson correlation method yields a substantially higher consistency percentage on this set: 71.1% as opposed to 48.5%. An inspection of the dimension orders shows that this is arguably the result of chance. The Pearson correlation and logistic regression methods agree on the signs of the coefficients of all dimensions except that of the `Distance` dimension. The Pearson correlation coefficient of this dimension is 0.001, while its coefficient from the logistic regression analysis is -0.01. We see that both methods assign a negligibly small value, which is because the `Distance` dimension violates the assumption that its values tend to favor either of the outcomes; if $x_{\texttt{Type}} = \text{in}$ then lower values of the `Distance` dimension are better for outcome 1, and if $x_{\texttt{Type}} = \text{out}$ then higher values are better for outcome 1. The Pearson correlation method happened to assign a small positive value to the coefficient, but it could have just as well produced a small negative coefficient for a slightly different sample; and the same holds for the logistic regression method.

What about the relation between the Φ_s formulas and Ψ? In fact, none of the possible inclusions hold, so its Euler diagram is the most general one, which is depicted in Figure 3a.

Remark 7. Note that the formulas involved in these situations can become very big: the Welfare dataset contains 12 features and 99,988 cases, so the forcing formula Φ_s will contain approximately $12 \cdot 99{,}988 \approx 1.2$ million atomic subformulas. Nevertheless, Z3 is capable of handling big formulas such as these.

Part of the analysis performed by Steging et al. (2021) used a simplified version of the welfare set containing only a subset of the features of the original set: namely `Sex`, `Age`, `Type`, and `Distance`. The labels of this set are determined only by conditions C_1 and C_6, i.e. its labelling formula Ψ is defined as:

$$\begin{aligned} \Psi &:= \bigwedge\nolimits_{1 \leq i \leq 6} C_i, \\ C_1 &:= (x_{\texttt{Sex}} = \text{F} \wedge x_{\texttt{Age}} \geq 60) \vee (x_{\texttt{Sex}} = \text{M} \wedge x_{\texttt{Age}} \geq 65), \\ C_6 &:= (x_{\texttt{Type}} = \text{in} \wedge x_{\texttt{Distance}} < 50) \vee (x_{\texttt{Type}} = \text{out} \wedge x_{\texttt{Distance}} \geq 50). \end{aligned} \tag{12}$$

We refer to this set as the simplified welfare dataset and performed a similar analysis on it as with the other sets. The results can be found in Table 2.

The consistency percentage on this dataset is not great—only about 67.3% for the Pearson correlation method, and only 66% for the logistic regression method. However, when we break down this percentage for both classes we see that the situation is more dire than it at first appears. The consistency percentage for class 0 is 86.4%, but that of class 1 is 0%. This is caused by a single landmark with label 0, which forces all cases with outcome 0 for outcome 1, which means that $\uparrow\mathcal{C}_1 \subseteq \downarrow\mathcal{C}_0$: any case forced for outcome 1 by the case base is also forced for outcome 0. Z3 can prove that the case base is complete, and so since $\uparrow\mathcal{C}_1 \subseteq \downarrow\mathcal{C}_0$ this means $\mathcal{F} \subseteq \downarrow\mathcal{C}_0$: all fact situations are forced for outcome 0 by the case base. Lastly, it can be shown that $\mathcal{F}_1 \subseteq \uparrow\mathcal{C}_1$. The Euler diagram corresponding to this situation is shown in Figure 10.

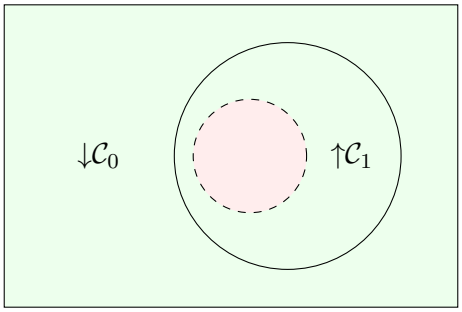


Figure 10: An Euler diagram representation of the relation between the simplified welfare case base and the ground truth labels determined by the formula in Eq. (12).

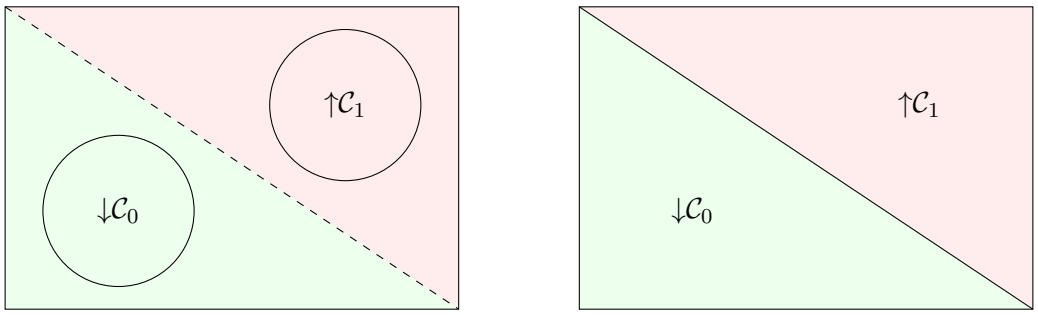
Why is the a fortiori model such a poor fit for this dataset? The reason, as mentioned previously, is that the features violate the assumption that their possible values have a preference for either of the binary outcomes. The way that `Distance` values prefer outcome 0 or 1 depends on the value of the `Type` dimension. Similarly, the `Type` and `Sex` dimensions do not themselves prefer outcome 0 or 1; they are just information to be conditioned on in the labelling formula. The only exception is the `Age` dimension, for which higher values clearly prefer outcome 1.

The simplified welfare set isolates almost exactly the variables that violate the dimension order assumption, which is also indicated by the fact that both the Pearson correlation and logistic regression methods assign coefficients to these dimensions which are very close to 0. What if we do the opposite: isolate from the original welfare dataset exactly the variables that satisfy the dimension order assumption? This means removing the `Distance`, `Type`, and `Sex` dimensions, and relabelling the data according to the following formula:

$$\begin{aligned} \Psi &:= \bigwedge\nolimits_{1\leq i\leq 5} C_i, \\ C_1 &:= x_{\texttt{Age}} \geq 60, \\ C_2 &:= 4 \leq \sum\nolimits_{1\leq i\leq 5} x_{\texttt{Con}_i}, \\ C_3 &:= x_{\texttt{Spouse}}, \\ C_4 &:= \neg x_{\texttt{Absent}}, \\ C_5 &:= x_{\texttt{Resources}} \leq 3{,}000. \end{aligned} \tag{13}$$

Performing this modification and subsequently removing duplicates yields a new dataset with 96,348 cases, which we will refer to as the second simplified welfare dataset. Fitting the a fortiori model on this set we get a consistency percentage of 100%. Moreover, Z3 can prove that $\mathcal{F}_0 \subseteq \downarrow\mathcal{C}_0$ and $\mathcal{F}_1 \subseteq \uparrow\mathcal{C}_1$. The Euler diagram corresponding to this situation is depicted in Figure 11a.

We see that the only property missing now is completeness. This means that it might be possible to add certain cases, so that the result is a consistent and complete case base. To finish this section on data analysis, we show that Z3 can potentially be used to complete a case base in such a scenario. This works because in order to prove completeness Z3 tries to find a counterexample, i.e. a fact situation $F \in \mathcal{F} \setminus (\downarrow\mathcal{C}_0 \cup \uparrow\mathcal{C}_1)$. If it succeeds at finding such a fact situation, we can determine its label using Eq. (13) and add it to the case base,



(a) Before completion (b) After completion

Figure 11: Euler diagram representations of the relation between the second simplified welfare case base and the ground truth labels determined by the formula in Eq. (13), before (11a) and after (11b) running Algorithm 1.

after which we ask Z3 to prove completeness again. This yields an algorithm for completing a case base, described in Algorithm 1. This algorithm does not necessarily terminate; e.g. Proposition 2 tells us it would loop endlessly on the CORELS dataset. If it does terminate, this is either because the case base was made inconsistent, or it was made complete by the addition of the last added case, while retaining the consistency property.

Algorithm 1: Completing a case base using Z3.

Data: A consistent, incomplete case base $\mathcal{C}$
while $\mathcal{C}$ *is consistent and incomplete* **do**
  $F \leftarrow$ the counterexample to completeness generated by Z3;
  $s \leftarrow$ the ground truth label of F according to the labelling formula Ψ;
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(F, s)\}$;
end

Running Algorithm 1 on the second simplified welfare case base yields a consistent and complete case base with 19 landmarks; see Figure 11b for the corresponding Euler Diagram visualization.

4. Conclusion and Discussion

In recent work by Prakken and Ratsma (2022) an XAI method was developed on the basis of the a fortiori model of precedential constraint developed by Horty (2011). In the present work, we connected the theory behind this model to order theory and logic, and added notions of landmarks and completeness. We then used this logical perspective to implement the model in Python using the SMT solver Z3 (de Moura & Bjørner, 2008).

This implementation was used to evaluate how suitable Horty's theory is to model the kind of data we might encounter in cases that require explainability methods. As an example of such a situation we chose the COMPAS data published by Angwin et al. (2016). We fitted the model on this data and some variations upon it, and analyzed the results in the sense that we measured their consistency percentages, and looked at the structure of the forcing

relation on cases. Through this analysis, and the use of the concept of landmark cases, we extrapolate from these results an informal characterization of the type of datasets that are fit to be described by the theory of precedential constraint. This characterization can be viewed mathematically, as consistency indicating the degree of linear separability of the data; or viewed semantically, as consistency indicating the degree to which the process generating the data respects precedence, or depends on a fortiori type reasoning.

Furthermore, we considered several datasets used in the work by Steging et al. (2021), to analyze the fit of the a fortiori model on datasets that come with known ground truth labels. This exemplified the importance of the assumption that the features can be interpreted as having a dimension order, which underlies the a fortiori model. It also demonstrates the usefulness of implementing the model in Z3, as this allows us to automatically prove properties about the model, which would be intractable to do by hand.

These results raise several questions which may be addressed in future research. Firstly, there is the question of the degree to which our results depend on the statistical methods we used to determine the dimension orders. In the framework developed by Horty (2019) the dimension orders may be partial, as opposed to linear, but the statistical methods we used can only produce linear orders. What are the situations in which we might want to make elements of a dimension incomparable, and how would the presence of incomparable pairs affect our findings regarding consistency? Closely related is a second question regarding the method of determining these orders. In this work, we used logistic regression, but other ways of doing this are conceivable. What are the differences between these approaches, and what should the measure of success be? Especially in a setting where the theory of precedential constraint underlies an explanation method (as is the case for the method developed by Prakken and Ratsma, 2022), it might be better to use the black box that is under examination to determine these orders, because in that case we are less interested in what we think the orders should be and more in what the black box thinks they should be. Thirdly, since the result model is in some sense itself a data-driven model, we can ask how it compares to other such models. The model on the CORELS data, shown in Figure 8, suggests it functions as a type of decision tree, but an obvious difference is that it is not always capable of classifying an arbitrary unseen case (if the case base in question is incomplete), and that it may give conflicting classifications (if it is inconsistent).

Acknowledgments

This article is an extended version of the HHAI2022 conference paper by van Woerkom et al. (2022). It includes additional material on the theoretical aspects of the a fortiori model, a new software implementation based on these theoretical results, and additional dataset evaluations. This research was (partially) funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, grant number 024.004.022.

Appendix A. Z3 Implementation Example

In this appendix we provide some code snippets of our implementation described in Section 3.1. This is a brief description intended to illustrate the close resemblance between the logical language of Section 2.3.2 and the Python code; to see the full implementation details we refer the reader to our GitHub page.[4] To start, let `D` be a list containing the names of the dimensions as strings, and `orders` a dictionary mapping `D` to appropriate Z3 sorts. For instance, for the `Age` dimension of our recidivism example `D` will contain a string `"age"`, and `orders["age"]` will return the Z3 sort for integers `IntSort()`.

We define the set of variables as a dictionary mapping the elements of `D` to Z3 constants:

```
x = {d : Const(f'x_{d}', sorts[d]) for d in D}
```

In Z3 there is no difference between constants and variables; the solver will try to find interpretations for uninterpreted constants appearing in the formulas it tries to satisfy.

Similarly, a fact situation `F` is represented as a dictionary mapping the elements of `D` to particular Z3 constants. A case `a` is a fact situation that additionally maps the string `"Label"` to either 0 or 1.

Next we use a logistic regression analysis on the data to construct a dictionary `orders` that maps elements of `D` to an order. For instance, for the `Age` dimension of our recidivism example `orders["age"]` will return the `operator.ge` object from the `operator` package, which act as the $\geq$ order when applied to integers.

It is now straightforward to define the forcing formulas Φ_s. To start we define $\phi_s(F)$ by:

```
def f(F, s, x):
  if s == 1:
      return And([orders[d](F[d], x[d]) for d in D])
  elif s == 0:
      return And([orders[d](x[d], F[d]) for d in D])
```

Using `f(F, s, x)` we define Φ_s as:

```
def F(C, s, x):
  return Or([f(a, s, x) for a in C if a["Label"] == s])
```

Now we can define, e.g., the formula $\Phi_0 \vee \Phi_1$ expressing completeness by:

```
compl = Or(F(C, 0, x), F(C, 1, x)).
```

To test whether the case base is complete we make a solver `s` and check validity of `compl` by checking unsatisfiability of `Not(compl)`:

```
s = Solver()
s.add(Not(compl))
s.check()
```

If $\mathcal{C}$ is complete then this code will return the `unsat` answer, and otherwise it will produce a model containing a witness to incompleteness—i.e. a fact situation satisfying $\neg\Phi_0 \wedge \neg\Phi_1$.

4. https://github.com/wijnanduu/AFCBR.

References

Acharya, M. S., Armaan, A., & Antony, A. S. (2019). A comparison of regression models for prediction of graduate admissions. *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, 1–5. https://doi.org/10.1109/ICCIDS.2019.8862140

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., & Rudin, C. (2018). Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, *18*(234), 1–78. http://jmlr.org/papers/v18/17-716.html

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. *ProPublica*. Retrieved February 26, 2024, from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Barenstein, M. (2019). *ProPublica's COMPAS data revisited*. arXiv: 1906.04711 [cs, econ, q-fin, stat]. https://doi.org/10.48550/arXiv.1906.04711

Bench-Capon, T. (1993). Neural networks and open texture. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, 292–297. https://doi.org/10.1145/158976.159012

Bjørner, N., & Nachmanson, L. (2020). Navigating the universe of Z3 theory solvers. *Formal Methods: Foundations and Applications*, 8–24. https://doi.org/10.1007/978-3-030-63882-5_2

Bradley, A. R., & Manna, Z. (2007). *The calculus of computation*. Springer. https://doi.org/10.1007/978-3-540-74113-8

Čyras, K., Satoh, K., & Toni, F. (2016). Explanation for case-based reasoning via abstract argumentation. In P. Baroni, T. F. Gordon, T. Scheffler, & M. Stede (Eds.), *Computational Models of Argument. Proceedings of COMMA 2016* (pp. 243–254). IOS Press. https://doi.org/10.3233/978-1-61499-686-6-243

Davey, B. A., & Priestley, H. A. (2002). *Introduction to lattices and order* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511809088

de Moura, L., & Bjørner, N. (2008). Z3: An efficient SMT solver. *Tools and Algorithms for the Construction and Analysis of Systems*, 337–340. https://doi.org/10.1007/978-3-540-78800-3_24

de Moura, L., & Bjørner, N. (2009). Satisfiability modulo theories: An appetizer. *Formal Methods: Foundations and Applications*, 23–36. https://doi.org/10.1007/978-3-642-10452-7_3

Dieterich, W., Mendoza, C., & Brennan, T. (2016). *COMPAS risk scales: Demonstrating accuracy equity and predictive parity* (Research report). Northpointe Inc. Research Department.

Horty, J. (2011). Rules and reasons in the theory of precedent. *Legal Theory*, *17*(1), 1–33. https://doi.org/10.1017/S1352325211000036

Horty, J. (2019). Reasoning with dimensions and magnitudes. *Artificial Intelligence and Law*, *27*(3), 309–345. https://doi.org/10.1007/s10506-019-09245-0

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. *Proceedings of the 34th International Conference on Machine Learning*, 1885–1894. https://proceedings.mlr.press/v70/koh17a.html

Kurita, K., Vyas, N., Pareek, A., Black, A. W., & Tsvetkov, Y. (2019). Measuring bias in contextualized word representations. *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, 166–172. https://doi.org/10.18653/v1/W19-3823

Larson, J., Mattu, S., Kirchner, L., & Angwin, J. (2016). *How we analyzed the COMPAS recidivism algorithm*. ProPublica. Retrieved May 16, 2024, from https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

Liu, X., Lorini, E., Rotolo, A., & Sartor, G. (2022). Modelling and explaining legal case-based reasoners through classifiers. In E. Francesconi, G. Borges, & C. Sorge (Eds.), *Legal Knowledge and Information Systems. JURIX 2022: The Thirty-fifth Annual Conference* (pp. 83–92). IOS Press. https://doi.org/10.3233/FAIA220451

Manzano, M., & Aranda, V. (2022). Many-sorted logic. In *The Stanford Encyclopedia of Philosophy* (Winter 2022). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2022/entries/logic-many-sorted/

Nugent, C., & Cunningham, P. (2005). A case-based explanation system for black-box systems. *Artificial Intelligence Review*, *24*(2), 163–178. https://doi.org/10.1007/s10462-005-4609-5

Odekerken, D., & Bex, F. (2020). Towards transparent human-in-the-loop classification of fraudulent web shops. In Serena Villata, Jakub Harašta, & Petr Křemen (Eds.), *Legal Knowledge and Information Systems. JURIX 2020: The Thirty-third Annual Conference* (pp. 239–242). IOS Press. https://doi.org/10.3233/FAIA200873

Odekerken, D., Bex, F., & Prakken, H. (2023). Justification, stability and relevance for case-based reasoning with incomplete focus cases. *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, 177–186. https://doi.org/10.1145/3594536.3595136

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Peters, J. G., Bex, F., & Prakken, H. (2022). Justifications derived from inconsistent case bases using authoritativeness. *1st International Workshop on Argumentation for eXplainable AI*, *3209*. https://ceur-ws.org/Vol-3209

Peters, J. G., Bex, F., & Prakken, H. (2023). Model- and data-agnostic justifications with a fortiori case-based argumentation. *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, 207–216. https://doi.org/10.1145/3594536.3595164

Prakken, H., & Ratsma, R. (2022). A top-level model of case-based argumentation for explanation: Formalisation and experiments. *Argument & Computation*, *13*(2), 159–194. https://doi.org/10.3233/AAC-210009

Prakken, H., & Sartor, G. (1998). Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, *6*(2), 231–287. https://doi.org/10.1023/A:1008278309945

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Demonstrations, 97–101. https://doi.org/10.18653/v1/N16-3020

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Rudin, C., Wang, C., & Coker, B. (2020). The age of secrecy and unfairness in recidivism prediction. *Harvard Data Science Review*, *2*(1). https://doi.org/10.1162/99608f92.6ed64b30

Schlimmer, J. (1981). *Mushroom dataset*. https://doi.org/10.24432/C5959T

Steging, C., Renooij, S., & Verheij, B. (2021). Discovering the rationale of decisions: Towards a method for aligning learning and reasoning. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 235–239. https://doi.org/10.1145/3462757.3466059

van Woerkom, W., Grossi, D., Prakken, H., & Verheij, B. (2022). Landmarks in case-based reasoning: From theory to data. *HHAI2022: Augmenting Human Intellect*, *354*, 212–224. https://doi.org/10.3233/FAIA220200

van Woerkom, W., Grossi, D., Prakken, H., & Verheij, B. (2023). Hierarchical precedential constraint. *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, 333–342. https://doi.org/10.1145/3594536.3595154

Verheij, B. (2017). Formalizing arguments, rules and cases. *Proceedings of the Sixteenth International Conference on Articial Intelligence and Law*, 199–208. https://doi.org/10.1145/3086512.3086533

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, *7*(2), 76–99. https://doi.org/10.1093/idpl/ipx005

Wachter, S., Mittelstadt, B., & Russell, C. (2018). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, *31*(2). https://doi.org/10.2139/ssrn.3063289