On Belief Change for Multi-Label Classifier Encodings

Sylvie Coste-Marquis[1] and **Pierre Marquis**[1,2]
[1]CRIL, Univ. Artois & CNRS, France
[2]Institut Universitaire de France, France
{coste, marquis}@cril.fr

Abstract

An important issue in ML consists in developing approaches exploiting background knowledge T for improving the accuracy and the robustness of learned classifiers $\boldsymbol{C}$. Delegating the classification task to a Boolean circuit Σ exhibiting the same input-output behaviour as $\boldsymbol{C}$, the problem of exploiting T within $\boldsymbol{C}$ can be viewed as a belief change scenario. However, usual change operations are not suited to the task of modifying the classifier encoding Σ in a minimal way, to make it complying with T. To fill the gap, we present a new belief change operation, called *rectification*. We characterize the family of rectification operators from an axiomatic perspective and exhibit operators from this family. We identify the standard belief change postulates that every rectification operator satisfies and those it does not. We also focus on some computational aspects of rectification and compliance.

1 Introduction

Integrating learning and reasoning is one of the key challenges of AI today, and as such, it attracted much attention from several scientific communities, especially the neural-symbolic computation one (NeSy) and the statistical relational learning and AI one (StarAI) [Raedt *et al.*, 2020; Besold *et al.*, 2017; d'Avila Garcez *et al.*, 2019; Russell, 2015; Raedt *et al.*, 2016]. The motivations for integrating learning and reasoning are numerous, giving rise to many research issues and associated methods for handling them.

One of these issues consists in leveraging knowledge to improve machine learning (ML) systems (especially, classifiers of various types) in terms of accuracy / data efficiency (see e.g., [Hu *et al.*, 2016; Donadello *et al.*, 2017; Xu *et al.*, 2018; Xie *et al.*, 2019; Chen *et al.*, 2020]). For instance, a background theory T can be exploited during the learning phase used to generate the parameters of $\boldsymbol{C}$; T can be viewed as a soft constraint that promotes solutions that are close to satisfying assignments, and this proves useful for generating better classifiers (see e.g., [Xie *et al.*, 2019]).

However, such approaches do not guarantee that the learned classifier will classify every new instance in a way that is consistent with T. Furthermore, they are not applicable when the classifier has already been learned and the training set is not available any longer. In such a case, the learned classifier $\boldsymbol{C}$ itself must be *modified* to comply with T.

Noticeably, recent works have shown how ML classifiers $\boldsymbol{C}$ of various types can be encoded as Boolean circuits Σ (alias transparent or "white" boxes) (see e.g., [Narodytska *et al.*, 2018; Shih *et al.*, 2019; Shi *et al.*, 2020]). In the following, we do not make any strong assumption on the nature of the classifier that is considered. $\boldsymbol{C}$ is supposed to be a discrete *multi-label classifier*: given a set $X = \{x_1, \cdots, x_n\}$ (its elements are Boolean features) and a set $Y = \{y_1, \cdots, y_m\}$, that is disjoint with X (its elements are the labels, denoting classes / concepts), $\boldsymbol{C}$ is a mapping associating with each input instance (a vector $\boldsymbol{x} \in \boldsymbol{X}$ of n Boolean values assigned to the variables of X) a vector $\boldsymbol{y} \in \boldsymbol{Y}$ of m Boolean values assigned to the variables of Y. We also assume that a Boolean circuit Σ that encodes $\boldsymbol{C}$ is available. Σ is required to have the same input-output behaviour as $\boldsymbol{C}$: for any pair $(\boldsymbol{x}, \boldsymbol{y})$ for which $\boldsymbol{y} = \boldsymbol{C}(\boldsymbol{x})$, we have $y_j = 1$ precisely when the output variable y_j of the circuit Σ on the input $\boldsymbol{x}$ is set to 1. Such a circuit Σ can be viewed as a compact representation of the classes Y, as they are recognized by $\boldsymbol{C}$.

Thanks to such encodings, the classification task w.r.t. $\boldsymbol{C}$ can be based on the associated circuit Σ, and achieved efficiently from this circuit. Indeed, it is well-known that the circuit value problem that consists of computing the output of a given Boolean circuit on a given input is complete for P under uniform AC^0 reductions.[1] In terms of time complexity, this problem can be solved in linear time in the size of the circuit simply by a topological sort. Furthermore, beyond the classification task, both explanation and verification queries about $\boldsymbol{C}$ can be addressed via the corresponding classification circuit Σ [Darwiche and Hirth, 2020; Audemard *et al.*, 2020].

Interestingly, it turns out that the problem of modifying $\boldsymbol{C}$ once learned so that it complies with T can also be delegated to the corresponding Boolean circuit Σ. This problem amounts to a *belief change issue*, a question studied for decades in the knowledge representation (KR) community. Here, T is supposed to be more reliable than Σ because of

[1] We assume the reader acquainted with basics of complexity theory (see e.g., [Papadimitriou, 1994] otherwise).

the noise that may pervade the data used to learn the classifier $\boldsymbol{C}$. One is then interested in modifying Σ as few as possible in order to comply with T. What does "comply" mean depends of the nature of change one wants to deal with (e.g., belief revision, belief update). However, as we shall show soon, usual belief change operations (belief revision and belief update) that gave rise to an abundant literature in KR are not suited to the task at hand. A new change operation dedicated to multi-label classifier encodings is needed.

In the rest of the paper, the following research questions are considered:

- How to characterize multi-label classifier encodings Σ?
- What does it mean for a multi-label classifier encoding Σ to comply with some background knowledge T?
- How to define belief change operators $\star$ that are suited to the task of making a multi-label classifier encoding Σ to comply with some background knowledge T?
- How do $\star$ operators differ from standard belief change operators, i.e., belief revision and belief update operators?
- How hard is it to compute rectified classifications, i.e., classifications from $\Sigma \star T$?
- How hard is it to evaluate how much a multi-label classifier encoding Σ complies with a background theory T?

Our contributions are as follows. A logical characterization of multi-label classifier encodings is pointed out. Several notions of compliance of multi-label classifier encodings with some background knowledge are presented. A new belief change operation, called *rectification*, that is specific to multi-label classifier encodings, is introduced. We characterize the family of rectification operators from an axiomatic perspective and exhibit some operators from this family. In addition, the standard belief change postulates that every rectification operator satisfies and those it does not are identified. Especially, we prove that the families of rectification operators and those of revision operators / update operators are disjoint. The problem of computing rectified classifications is shown hard in general, but tractable restrictions of this problem are provided. Similarly, we show that evaluating how much a multi-label classifier encoding complies with a background theory is computationally demanding in the general case, but easy when some requirements on the representations of Σ and T are satisfied.

The rest of the paper is organized as follows. After some formal preliminaries (Section 2), we present in Section 3 a condition, called XY-classification property, that characterizes multi-label classifier encodings; we present as well several notions of compliance suited to multi-label classifier encodings. In Section 4, the family of rectification operators is defined and its connections to other belief change operators are investigated. In Section 5, some computational issues about rectification and compliance are considered. Section 6 discusses additional related work. Finally, Section 7 concludes the paper and gives some perspectives for further research. Proofs are not reported in the paper for space reasons (a full-proof version of the paper is available at www.cril.fr/~marquis/rectification.pdf).

2 Formal Preliminaries

Propositional logic. The propositional languages $\mathcal{L}$ considered in this paper are defined over a finite and non-empty set PS of propositional variables (that includes but does not necessarily restrict to $X \cup Y$) and standard connectives. The elements of a propositional language $\mathcal{L}$ are called *representations*, and for any such representation Σ, we denote by $Var(\Sigma)$ the subset of variables of PS occurring in Σ. As usual, atomic representations include propositional variables in PS, and Boolean constants in $\{\top, \bot\}$. A *literal* is a propositional variable, possibly negated, or a Boolean constant. Any propositional variable x is called a *positive literal*, and the negation of x, denoted $\neg x$ or $\overline{x}$, is called a *negative literal*. If ℓ is a literal x (resp. $\neg x$), then its complementary literal $\sim \ell$ is $\neg x$ (resp. x). For any subset X of PS, L_X denotes the set of literals based on the variables of X. A *term* is a conjunction of literals, and a *clause* is a disjunction of literals. A *canonical term* over X is a consistent term into which every variable of X occurs (as such, or negated).

Given a set of variables $V \subseteq PS$, an *interpretation over* V is a mapping ω from V to $\mathbb{B} = \{0, 1\}$. When a total ordering $<$ over PS is provided, interpretations can be represented by bit vectors from the set $\boldsymbol{V}$. For instance, if $V = \{v_1, v_2\}$ such that $v_1 < v_2$, then the mapping ω such that $\omega(v_1) = 0$ and $\omega(v_2) = 1$ can be represented by $(0, 1)$. Propositional representations are interpreted in a classical way. For a representation Σ and an interpretation over any superset of $V = Var(\Sigma)$, we use $\omega \models \Sigma$ to denote the fact that ω if a model of Σ according to the semantics of propositional logic. That is, assigning the variables of Σ to truth values as specified by ω makes Σ true. By $[\Sigma]$ we denote the *set* of models of Σ over $Var(\Sigma)$, and by $\|\Sigma\|$ we denote the *number* of models of Σ over $Var(\Sigma)$. In particular, Σ is *inconsistent* if $\|\Sigma\| = 0$, and *consistent* otherwise. Σ is said to be *complete* when it has a unique model. A representation Σ_2 is a *logical consequence* of a representation Σ_1 (denoted $\Sigma_1 \models \Sigma_2$) if $\Sigma_1 \wedge \neg\Sigma_2$ is inconsistent. Σ_1 and Σ_2 are *logically equivalent* (denoted $\Sigma_1 \equiv \Sigma_2$) if they are logical consequences of each other.

Given a representation Σ and a consistent term γ, the *conditioning* of Σ by γ is the representation obtained by replacing in Σ every occurrence of a variable $v \in Var(\gamma)$ by $\top$ if v is a positive literal of γ and by $\bot$ if $\neg v$ is a negative literal of γ. When V is a subset of propositional variables from PS, Σ is said to be *independent* of V if there is a representation Φ logically equivalent to Σ such that $Var(\Phi) \cap V = \varnothing$. The *forgetting* of V in Σ, denoted $\exists V.\Sigma$, is (up to logical equivalence) the *most general consequence* of Σ that is independent of V. The *projection* of Σ onto V is the forgetting of $\overline{V}$ in Σ, where $\overline{V}$ denotes the set $PS \setminus V$. Let us mention that $\exists V.\Sigma$ can be computed as a propositional representation, thanks to the following inductive characterization:

- $\exists \varnothing.\Sigma \equiv \Sigma$,
- $\exists \{v\}.\Sigma \equiv (\Sigma \mid \neg v) \vee (\Sigma \mid v)$,
- $\exists (V' \cup \{v\}).\Sigma \equiv \exists V'.(\exists \{v\}.\Sigma)$.

Finally, we are interested in properties of propositional languages $\mathcal{L}$ that state the existence of polynomial-time algorithms for answering some queries (consistency testing $\mathbf{CO}$, model counting $\mathbf{CT}$) from representations in $\mathcal{L}$, or for achieving some transformations over such representations (conditioning $\mathbf{CD}$, forgetting $\mathbf{FO}$, bounded conjunction $\wedge\mathbf{BC}$). See [Darwiche and Marquis, 2002] for details.

Propositional belief change. Belief revision consists in incorporating into an existing belief base φ (a propositional formula) a new piece of evidence μ (a propositional formula).

The following postulates have been pointed out for characterizing rational revision operators $\circ$ over a finite propositional language [Katsuno and Mendelzon, 1991b]:

Definition 1 (KM revision operator). *A* KM revision operator $\circ$ *is a mapping associating with a change formula $\mu \in \mathcal{L}$ and a formula $\varphi \in \mathcal{L}$, a new base $\varphi \circ \mu$ from $\mathcal{L}$, such that for every formula $\mu, \mu' \in \mathcal{L}$, for any consistent formulae $\varphi, \varphi' \in \mathcal{L}$, it satisfies the following postulates:*

(R1) $\varphi \circ \mu \models \mu$;

(R2) *If $\varphi \wedge \mu$ is consistent, then $\varphi \circ \mu \equiv \varphi \wedge \mu$;*

(R3) *If μ is consistent, then $\varphi \circ \mu$ is consistent;*

(R4) *If $\varphi \equiv \varphi'$ and $\mu \equiv \mu'$, then $\varphi \circ \mu \equiv \varphi' \circ \mu'$;*

(R5) $(\varphi \circ \mu) \wedge \mu' \models \varphi \circ (\mu \wedge \mu')$;

(R6) *If $(\varphi \circ \mu) \wedge \mu'$ is consistent,*
then $\varphi \circ (\mu \wedge \mu') \models (\varphi \circ \mu) \wedge \mu'$.

Belief update mainly focuses on determining how a belief state (typically represented by a belief base ψ) should evolve in order to take into account a new piece of information μ reflecting an explicit evolution of the world.

The following postulates [Katsuno and Mendelzon, 1991a] have been pointed out for characterizing rational update operators $\diamond$:

Definition 2 (KM update operator). *A* KM update operator $\diamond$ *is a mapping associating with a change formula $\mu \in \mathcal{L}$ and a formula $\psi \in \mathcal{L}$, a new base $\psi \diamond \mu$ from $\mathcal{L}$, such that for every formula $\mu, \mu' \in \mathcal{L}$, for any consistent formulae $\psi, \psi' \in \mathcal{L}$, it satisfies the following postulates:*

(U1) $\psi \diamond \mu \models \mu$;

(U2) *If $\psi \models \mu$ then $\psi \diamond \mu \equiv \psi$;*

(U3) *If ψ and μ are consistent then $\psi \diamond \mu$ is consistent;*

(U4) *If $\psi \equiv \psi'$ and $\mu' \equiv \mu'$ then $\psi \diamond \mu \equiv \psi' \diamond \mu'$;*

(U5) $(\psi \diamond \mu) \wedge \mu' \models \psi \diamond (\mu \wedge \mu')$;

(U6) *If $\psi \diamond \mu \models \mu'$ and $\psi \diamond \mu' \models \mu$ then $\psi \diamond \mu \equiv \psi \diamond \mu'$;*

(U7) *If ψ is complete then $(\psi \diamond \mu) \wedge (\psi \diamond \mu') \models \psi \diamond (\mu \vee \mu')$;*

(U8) *$(\psi \vee \psi') \diamond \mu$ is equivalent to $(\psi \diamond \mu) \vee (\psi' \diamond \mu)$.*

3 Classification and Compliance

Let us first make formal what "classifying" means when dealing with (general) propositional representations:

Definition 3 (classification)**.** *Let φ be a propositional representation in $\mathcal{L}$ over PS, and let X and Y be two disjoint subsets of PS. Let $\boldsymbol{x} \in \boldsymbol{X}$ and $\boldsymbol{y} \in \boldsymbol{Y}$. φ is said to* classify *$\boldsymbol{x}$ (as $\boldsymbol{y}$) if and only if the propositional representation $\varphi(\boldsymbol{x})$, defined by $\varphi(\boldsymbol{x}) = \exists\overline{Y}.(\varphi \mid \boldsymbol{x})$, has a unique model over Y, namely $\boldsymbol{y}$. The set of instances $\boldsymbol{x}$ that are* classified *by φ is $C(\varphi) = \{\boldsymbol{x} \in \boldsymbol{X} \mid \exists \boldsymbol{y} \in \boldsymbol{Y}$ s.t. φ classifies $\boldsymbol{x}$ as $\boldsymbol{y}\}$.*

Example 1. *Let $X = \{x_1, x_2\}$, $Y = \{y\}$, and $Z = \{z\}$. Let $\varphi = (x_1 \wedge x_2 \wedge y \wedge z) \vee (\overline{x_1} \wedge \overline{y})$. When $\boldsymbol{x} = (1, 1)$ we have $\varphi(\boldsymbol{x}) \equiv y$. Thus φ classifies $(1, 1)$ as (1). Similarly, φ classifies $(0, 0)$ and $(0, 1)$ as (0). On the other hand, when $\boldsymbol{x} = (1, 0)$ we have $\varphi(\boldsymbol{x}) \equiv \bot$, showing that φ does not classify $(1, 0)$. We thus have $C(\varphi) = \{(1, 1), (0, 0), (0, 1)\}$.*

For any propositional representation $\varphi \in \mathcal{L}$ and any $\boldsymbol{x} \in \boldsymbol{X}$, it can be observed that $\varphi(\boldsymbol{x}) \equiv (\exists\overline{X \cup Y}.\varphi)(\boldsymbol{x})$. Intuitively, this equivalence indicates that the pieces of information in φ that do not only depend on X and Y are irrelevant to the classification task when X represents the features and Y the labels.

Now, in order to be considered as the encoding of a Boolean classifier $\boldsymbol{C}$ with features in X and labels in Y, a propositional representation Σ is expected to exhibit the *XY-classification property*:

Definition 4 (XY-classification property)**.** *A propositional representation $\Sigma \in \mathcal{L}$ has the XY-*classification property *when it is equivalent to an XY-*classification circuit*, i.e., a DAG D such that:*

- *the source nodes of D are associated with the variables in X and for every variable y of Y, there is a sink node of D that is associated with y;*
- *every node N in D, except the source nodes, is associated with a specific variable v_N from $Y \cup Z$; v_N is the output of a gate, the inputs of it being the parents of N in D; this gate corresponds to an equivalence of the form $v_N \Leftrightarrow \varphi_N$, where φ_N is a formula from $\mathcal{L}$ over the variables associated with the parents of N.*

From a logical standpoint, an XY-classification circuit is viewed as the conjunction of all the equivalences associated with its nodes, thus a formula over $X \cup Y \cup Z$ where $X, Y, Z \subseteq PS$.

Example 2. *Let $X = \{x\}$, $Y = \{y\}$, and $Z = \{z\}$. Let $\Sigma = (x \Leftrightarrow z) \wedge (y \Leftrightarrow z)$. Obviously enough, Σ has the XY-classification property.*

When Σ has the XY-classification property, every variable v of Σ is *definable* in Σ in terms of X, meaning that there exists a formula φ_v^X built upon variables of X, only, such that $\Sigma \models v \Leftrightarrow \varphi_v^X$. Though necessary, this definability condition on the variables is not sufficient to characterize the XY-classification property. Indeed, any inconsistent representation Σ satisfies it, but an inconsistent Σ is not equivalent to any XY-classification circuit. Actually, the right concept required here is the one of *unambiguous definability* [Lang and Marquis, 2008]. Thus, beyond the definability condition, a propositional representation Σ that has the XY-classification property must be such that $\Sigma \mid \boldsymbol{x}$ is consistent for every $\boldsymbol{x} \in \boldsymbol{X}$ (this is mandatory to ensure that every instance $\boldsymbol{x}$ is associated with at least one $\boldsymbol{y} \in \boldsymbol{Y}$, the definability condition guaranteeing the unicity of $\boldsymbol{y}$).

Obviously enough, every propositional representation Σ that has the XY-classification property is such that $C(\Sigma) = \boldsymbol{X}$: Σ must classify every instance. Note that the converse implication does not hold: take $X = \{x\}$, $Y = \{y\}$, and

$Z = \{z\}$, and $\varphi = (x \wedge y \wedge z) \vee (\overline{x} \wedge \overline{y})$. φ is such that $C(\varphi) = \boldsymbol{X}$ but φ does not have the XY-classification property (z is not defined in φ in terms of X).

Assuming that a background theory T is available, it is expected that Σ complies with T. Indeed, T contains pieces of belief that can be supposed more certain than those implicitly recorded in Σ since the latter ones have been learned from data that can be noisy. Accordingly, the primacy is given to T, and when Σ does not comply with T, Σ should be (minimally) modified so as to become compliant with T. Depending on what "compliant" means, several belief change operations can be considered.

Thus, in the belief revision case, Σ complies with T precisely when Σ is consistent with T. When this is the case, the change operation that is expected simply consists in conjoining Σ with T, so as to retain all the models of T which are also models of Σ. This is imposed by the rationality postulate **(R2)** for revision. In the remaining case, some models of T are selected, the selection process being based on Σ.

It turns out that the notion of compliance and the corresponding change operation required for incorporating a background theory T into a XY-classification circuit Σ are not the ones considered in belief revision:

Example 3. *Let* $X = \{x_1, x_2\}$ *and* $Y = \{y\}$. *Let* $\Sigma = (x_1 \wedge x_2) \Leftrightarrow y$, $T = (x_1 \wedge \overline{x_2}) \Leftrightarrow y$. $\Sigma \wedge T$ *is consistent (it is equivalent to* $\overline{x_1} \wedge \overline{y}$*), thus following* ***(R2)****, a rational revision of* Σ *by* T *should be equivalent to* $\overline{x_1} \wedge \overline{y}$. *However,* $\overline{x_1} \wedge \overline{y}$ *is not equivalent to any* XY*-classification circuit.*

The problem is that the selection process achieved by the revision operation is performed *modelwise*. Indeed, the representation theorem for belief revision given in [Katsuno and Mendelzon, 1991b] indicates that the models of Σ revised by T are among the models of T that are minimal w.r.t. a faithful ordering associated with Σ. If one wants to guarantee that the resulting representation is equivalent to a XY-classification circuit, the selection process must satisfy a constraint that concerns the *whole set* of resulting models (and not each model taken individually). This constraint is given by the XY-classification property stating that, after the change, for every $\boldsymbol{x} \in \boldsymbol{X}$ there is a unique model extending $\boldsymbol{x}$ (this model also extends a unique $\boldsymbol{y} \in \boldsymbol{Y}$). Noticeably, instead of the standard notion of revision, considering refined revision operators as proposed in [Creignou *et al.*, 2014] for handling propositional fragments would not be enough to guarantee that Σ once revised by T has the XY-classification property.

Accordingly, the notion of compliance considered in belief revision (i.e., recovering consistency) is not convenient here. Several notions of compliance of an XY-classification circuit with a background theory T can be considered instead:

Definition 5 (compliances)**.** *Let* Σ, T *be two propositional representations from* $\mathcal{L}$ *s.t.* Σ *has the* XY*-classification property. Let* $\boldsymbol{x} \in \boldsymbol{X}$.

- Σ *is* classification-compliant *with* T *on* $\boldsymbol{x}$ *iff* $\Sigma(\boldsymbol{x}) \equiv T(\boldsymbol{x})$.
- Σ *is* knowledge-compliant *with* T *on* $\boldsymbol{x}$ *iff* $\Sigma(\boldsymbol{x}) \models T(\boldsymbol{x})$.
- Σ *is* fact-compliant *with* T *on* $\boldsymbol{x}$ *iff* $\Sigma(\boldsymbol{x}) \models F(T, \boldsymbol{x})$

where $F(T, \boldsymbol{x})$ $= \top$ *if* $T \mid \boldsymbol{x} \models \bot$,
$= \bigwedge_{\ell \in L_Y \text{ s.t. } T|\boldsymbol{x} \models \ell} \ell$ *otherwise.*

It can be easily checked that for every T *and every* $\boldsymbol{x}$*, we have* $T(\boldsymbol{x}) \models F(T, \boldsymbol{x})$. *In addition, for every* $\boldsymbol{x}$ *that is classified by* T*, we have* $T(\boldsymbol{x}) \equiv F(T, \boldsymbol{x})$.

Clearly enough, when Σ is classification-compliant with T on $\boldsymbol{x}$, we have $\boldsymbol{x} \in C(T)$: Σ and T classify $\boldsymbol{x}$ in the same way. Knowledge compliance is less demanding since it requires only that the classification achieved by $\boldsymbol{C}$ (thus, Σ) on $\boldsymbol{x}$ coheres with what T "says" about the classes of $\boldsymbol{x}$. Especially, knowledge compliance does not ask T to classify $\boldsymbol{x}$. Fact compliance asks even less since it does not focus on all what T implies about the classes of $\boldsymbol{x}$, but only on the class membership (or non-membership) relations that can be deduced from T. Thus, we have the following proposition:

Proposition 1. *Let* Σ, T *be two propositional representations from* $\mathcal{L}$ *s.t.* Σ *has the* XY*-classification property. Let* $\boldsymbol{x} \in \boldsymbol{X}$.

- *If* Σ *is classification-compliant with* T *on* $\boldsymbol{x}$*, then* Σ *is knowledge-compliant with* T *on* $\boldsymbol{x}$.
- *If* Σ *is knowledge-compliant with* T *on* $\boldsymbol{x}$*, then* Σ *is fact-compliant with* T *on* $\boldsymbol{x}$.

The converse implications do not hold:

Example 4. *Let* $X = \{x_1, x_2\}$ *and* $Y = \{y_1, y_2\}$. *Let* $\Sigma = (x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2)$ *and* $T = ((x_1 \wedge x_2) \Rightarrow (y_1 \wedge y_2)) \wedge (x_1 \Rightarrow (y_1 \vee y_2)) \wedge (\overline{x_1} \Rightarrow (y_1 \vee \overline{y_2}))$. *Let* $\boldsymbol{x}_1 = (1, 1)$, $\boldsymbol{x}_2 = (1, 0)$, $\boldsymbol{x}_3 = (0, 1)$. *We have:*

- $T(\boldsymbol{x}_1) \equiv y_1 \wedge y_2$*:* Σ *is classification-compliant with* T *on* $\boldsymbol{x}_1$ *since* $\Sigma(\boldsymbol{x}_1) \equiv y_1 \wedge y_2$.
- $T(\boldsymbol{x}_2) \equiv y_1 \vee y_2$*:* Σ *is knowledge-compliant with* T *on* $\boldsymbol{x}_2$*, but not classification-compliant with* T *on* $\boldsymbol{x}_2$*, since* $\Sigma(\boldsymbol{x}_2) \equiv y_1 \wedge \overline{y_2}$.
- $F(T, \boldsymbol{x}_3)$ *is valid, therefore* Σ *is fact-compliant with* T *on* $\boldsymbol{x}_3$ *but it is not knowledge-compliant with* T *on* $\boldsymbol{x}_3$*, given that* $T(\boldsymbol{x}_3) \equiv y_1 \vee \overline{y_2}$ *and* $\Sigma(\boldsymbol{x}_3) \equiv \overline{y_1} \wedge y_2$.

Observe that when T is inconsistent with $\boldsymbol{x}$, any representation Σ that has the XY-classification property is fact-compliant with T on $\boldsymbol{x}$ because $F(T, \boldsymbol{x})$ is valid. However, since $T(\boldsymbol{x})$ is inconsistent, Σ is neither knowledge-compliant nor classification-compliant with T on $\boldsymbol{x}$.

4 Rectifying a Classifier Representation

Whatever the notion of compliance under consideration, if Σ is not compliant with T on a given $\boldsymbol{x}$, then Σ must be "rectified" so as to make it compliant with T on $\boldsymbol{x}$ since primacy is given to T. Making it more formal calls for a notion of rectification operator:

Definition 6 (**rectification operator**)**.** *A* rectification operator $\star$ *is a mapping associating with two propositional representations* T *and* Σ *where* Σ *has the* XY*-classification property, a propositional representation* $\Sigma \star T$*, called a* rectified representation*, such that:*

(RE1) $\Sigma \star T$ *has the* XY*-classification property;*

(RE2) *If* Σ *is fact-compliant with* T *on* $\boldsymbol{x} \in \boldsymbol{X}$*, then* $(\Sigma \star T)(\boldsymbol{x}) \equiv \Sigma(\boldsymbol{x})$*;*

(RE3) *For any* $\boldsymbol{x} \in \boldsymbol{X}$, $(\Sigma \star T)(\boldsymbol{x}) \models F(T, \boldsymbol{x})$*;*

(RE4) *If T is inconsistent, then $\Sigma \star T \equiv \Sigma$;*

(RE5) *If $\Sigma \equiv \Sigma'$ and $T \equiv T'$, then $\Sigma \star T \equiv \Sigma' \star T'$;*

(RE6) $\Sigma \star T \equiv (\exists \overline{X \cup Y}.\Sigma) \star (\exists \overline{X \cup Y}.T)$.

The admissible mappings $\star$ are required to satisfy a number of postulates, namely the rectification postulates **(RE1-RE6)**. **(RE1)** states a key condition, asking any rectification operation to preserve the XY-classification property. It is the most demanding postulate since it requires that for every $\boldsymbol{x} \in \boldsymbol{X}$, $\Sigma \star T$ is consistent with $\boldsymbol{x}$ and furthermore that $(\Sigma \star T) \wedge \boldsymbol{x}$ has a single model. **(RE2)**, **(RE3)**, and **(RE4)** capture the principle of minimal change and the primacy of T over Σ. Thus, **(RE2)** states that there is no reason to rectify the classification of any $\boldsymbol{x}$ as achieved by Σ whenever Σ is fact-compliant with T on $\boldsymbol{x}$ when Σ is classification-compliant with T on $\boldsymbol{x}$, as shown by Proposition 1. In the remaining case, **(RE3)** constrains the way the rectified representation $\Sigma \star T$ classifies $\boldsymbol{x}$ by requiring that the class membership relationships imposed by T on $\boldsymbol{x}$ are satisfied. Indeed, **(RE3)** ensures that $\Sigma \star T$ is fact-compliant with T on every $\boldsymbol{x}$. For those $\boldsymbol{x}$ that are classified by T, **(RE3)** ensures that $\Sigma \star T$ is knowledge-compliant with T on $\boldsymbol{x}$; more than that, **(RE1)** and **(RE3)** together guarantee that $\Sigma \star T$ is classification-compliant with T on every $\boldsymbol{x}$ that is classified by T. Note that $F(T, \boldsymbol{x})$ is valid whenever T is inconsistent so that **(RE3)** is trivially satisfied in this case. **(RE4)** deals with the case when T is inconsistent; in such a situation, a minimal change of Σ consists in not modifying it at all. **(RE5)** is a (rather standard) syntax-independence postulate (it has the same form as the ones of **(R4)** and **(U4)**). Finally, **(RE6)** states that the result of rectifying Σ by T must not depend on the variables not directly involved in the classification task, i.e., those variables outside $X \cup Y$.

At that stage, it is important is to check that one can find operators $\star$ that do satisfy **(RE1-RE6)**:

Definition 7 ($\star_D$, $\star^s$)**.**

- *Let $\circ_D$ denote Dalal revision operator [Dalal, 1988], such that for any φ, α, the models of $\varphi \circ_D \alpha$ consist of the models of α which are as close as possible to φ w.r.t. Hamming distance. Let $\star_D$ be the mapping associating with a background formula T and a propositional representation Σ that has the XY-classification property, a propositional representation $\Sigma \star_D T$ such that*

$$\Sigma \star_D T \equiv \bigvee_{\boldsymbol{x} \in \boldsymbol{X}} \boldsymbol{x} \wedge (\Sigma \star_D T)(\boldsymbol{x})$$

where for any $\boldsymbol{x} \in \boldsymbol{X}$, $(\Sigma \star_D T)(\boldsymbol{x}) = \Sigma(\boldsymbol{x}) \circ_D F(T, \boldsymbol{x})$.

- *Let s be any selection function, i.e., a mapping associating with a consistent propositional representation α over Y a canonical term $s(\alpha)$ representing a model of α. We assume that s is syntax-independent, meaning that $s(\alpha) = s(\alpha')$ whenever $\alpha \equiv \alpha'$.*

  Let $\star^s$ be the mapping associating with a background theory T and a propositional representation Σ that has the XY-classification property, a propositional representation $\Sigma \star^s T$ such that

$$\Sigma \star^s T \equiv \bigvee_{\boldsymbol{x} \in \boldsymbol{X}} \boldsymbol{x} \wedge (\Sigma \star^s T)(\boldsymbol{x})$$

where for any $\boldsymbol{x} \in \boldsymbol{X}$,

- *$(\Sigma \star^s T)(\boldsymbol{x}) = \Sigma(\boldsymbol{x})$ if Σ is fact-compliant with T on $\boldsymbol{x}$,*
- *$(\Sigma \star^s T)(\boldsymbol{x}) = s(T(\boldsymbol{x}))$ otherwise.*

Example 5. *Let us consider again the sets of variables $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2\}$, and the formula $\Sigma = (x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2)$ given at Example 4, and let us take now $T = (x_1 \wedge x_2 \wedge y_1 \wedge y_2) \vee (\overline{x_1} \wedge y_1)$. We have $T((0,1)) \equiv F(T,(0,1)) \equiv y_1$. Since $\Sigma((0,1)) \equiv \overline{y_1} \wedge y_2$, Σ is not fact-compliant with T on $(0,1)$. We have $(\Sigma \star_D T)((0,1)) \equiv (\overline{y_1} \wedge y_2) \circ_D y_1 \equiv y_1 \wedge y_2$. If for any $\boldsymbol{x}$, $s(T(\boldsymbol{x}))$ is given as the minimal element of $T(\boldsymbol{x})$ w.r.t. the ordering $<$ over the interpretations over Y such that $00 < 01 < 10 < 11$, we have $(\Sigma \star^s T)((0,1)) \equiv s(y_1) \equiv y_1 \wedge \overline{y_2}$. Hence, we have $\Sigma \star^s T \not\equiv \Sigma \star_D T$.*

Observe that $\star^s$ is well-defined since for any $\boldsymbol{x}$ such that $T(\boldsymbol{x})$ is inconsistent, Σ is fact-compliant with T on $\boldsymbol{x}$.

Proposition 2. *$\star_D$ and every $\star^s$ are rectification operators.*

Taking advantage of $\star_D$ for rectifying Σ so as to account for T simply consists, for every $\boldsymbol{x} \in \boldsymbol{X}$, in revising $\Sigma(\boldsymbol{x})$ by $F(T, \boldsymbol{x})$ using Dalal revision operator. Since $\Sigma(\boldsymbol{x})$ and $F(T, \boldsymbol{x})$ are terms over Y, the revision process enforces every literal of $F(T, \boldsymbol{x})$ to hold in $(\Sigma \star_D T)(\boldsymbol{x})$ while keeping unchanged every other literal of $\Sigma(\boldsymbol{x})$ (thus, ensuring that $(\Sigma \star_D T)(\boldsymbol{x})$ has a single model over Y). As expected, the revision step using $\star_D$ has no effect on $\Sigma(\boldsymbol{x})$ whenever Σ is fact-compliant with T on $\boldsymbol{x}$.

A similar conclusion can be drawn when considering any $\star^s$ instead of $\star_D$. However, for those $\boldsymbol{x}$ such that Σ is not fact-compliant with T on $\boldsymbol{x}$, when $\star^s$ is considered, a more severe revision of $\Sigma(\boldsymbol{x})$ is achieved; indeed, $s(T(\boldsymbol{x}))$ implies $F(T, \boldsymbol{x})$ but the converse does not hold in general. Since $s(T(\boldsymbol{x})) \models T(\boldsymbol{x})$ holds, it is guaranteed that $\Sigma \star^s T$ is knowledge-compliant with T on such $\boldsymbol{x}$ (note that this property of knowledge-compliance is not guaranteed for those $\boldsymbol{x}$ when $\star_D$ is used instead). Clearly, the choice of a $\star^s$ operator comes at the expense of specifying a selection function s, but in many cases one can take advantage of additional information to define s. For instance, one can select a model of $T(\boldsymbol{x})$ which is implied by a maximal number of models of Σ (i.e., a most frequent class assignment).

Note that when dealing with mono-label classification problems, $\star_D$ and $\star^s$ coincide, whatever s:

Proposition 3. *If $|Y| = 1$, then $\star^s = \star_D$ for every s.*

As explained previously, the **(RE3)** postulate asks the rectified representation $\Sigma \star T$ to satisfy the less demanding form of compliance, namely fact-compliance. Two strengthenings of the **(RE3)** postulate, suited respectively to knowledge-compliance and classification-compliance, could have been envisioned instead:

(RE3') For any $\boldsymbol{x} \in \boldsymbol{X}$, $(\Sigma \star T)(\boldsymbol{x}) \models T(\boldsymbol{x})$;

(RE3") For any $\boldsymbol{x} \in \boldsymbol{X}$, $(\Sigma \star T)(\boldsymbol{x}) \models T(\boldsymbol{x})$, and for any $\boldsymbol{x} \in \boldsymbol{X}$ that is classified by T, we have

$$(\Sigma \star T)(\boldsymbol{x}) \equiv T(\boldsymbol{x}).$$

(RE3') strengthens **(RE3)** in order to ensure that the rectified representation $\Sigma \star T$ is knowledge-compliant with T on every x, and **(RE3")** strengthens **(RE3')** in order to ensure in addition that the rectified representation $\Sigma \star T$ is classification-compliant with T on every $\boldsymbol{x}$ that is classified by T. It can be observed that stating **(RE3")** as $\forall \mathbf{x} \in \mathbf{X}$, $(\Sigma \star T)(\mathbf{x}) \equiv T(\mathbf{x})$ would not be appropriate because this statement conflicts with **(RE1)** given that T is not required to satisfy the XY-classification property. Since the condition stating that for any $\boldsymbol{x} \in \boldsymbol{X}$ that is classified by T, $(\Sigma \star T)(\boldsymbol{x}) \equiv T(\boldsymbol{x})$ holds is already satisfied by the rectification operators in the sense of Definition 6, **(RE3")** actually is equivalent to **(RE3')** in presence of the other postulates. Thus considering **(RE3")** in addition to **(RE1-RE6)** and **(RE3')** would be useless.

Finally, **(RE3')** has not been retained because this postulate is incompatible with **(RE2)**. Indeed, it can be the case that $\Sigma(\boldsymbol{x}) \models F(T, \boldsymbol{x})$ while $\Sigma(\boldsymbol{x}) \not\models T(\boldsymbol{x})$. Take for instance $X = \{x\}$, $Y = \{y_1, y_2\}$, $\Sigma = (x \Leftrightarrow y_1) \wedge (x \Leftrightarrow y_2)$, and $T = x \Leftrightarrow (\overline{y_1} \vee \overline{y_2})$. For $\boldsymbol{x} = (1)$, we have $\Sigma(\boldsymbol{x}) \equiv y_1 \wedge y_2$, $F(T, \boldsymbol{x}) \equiv \top$, and $T(\boldsymbol{x}) \equiv \overline{y_1} \vee \overline{y_2}$. Thus, no rectification operator can satisfy **(RE3')**.

Nevertheless, it can be noted that **(RE3')** is compatible with the following weakening **(RE2')** of **(RE2)**:

(RE2') If Σ is knowledge-compliant with T on $\boldsymbol{x} \in \boldsymbol{X}$, then $(\Sigma \star T)(\boldsymbol{x}) \equiv \Sigma(\boldsymbol{x})$.

Accordingly, we can define a family of operators $*^s$ (similar to $\star^s$) as follows. Let s be any selection function, as in Definition 7. We define the $*^s$ operator by:

$$\Sigma *^s T \equiv \bigvee_{\boldsymbol{x} \in \boldsymbol{X}} \boldsymbol{x} \wedge (\Sigma *^s T)(\boldsymbol{x})$$

where for any $\boldsymbol{x} \in \boldsymbol{X}$,

- $(\Sigma *^s T)(\boldsymbol{x}) = \Sigma(\boldsymbol{x})$ if Σ is knowledge-compliant with T on $\boldsymbol{x}$,
- $(\Sigma *^s T)(\boldsymbol{x}) = s(T(\boldsymbol{x}))$ otherwise.

It can be shown that $*^s$ operators satisfy **(RE2')** and **(RE3')** (for space reasons, we refrain from presenting $*^s$ operators in more detail in this paper).

Finally, in order to figure out in a more accurate way how rectification operators differ from other change operators, we have evaluated the compatibility of the rectification postulates with those characterizing belief revision operators and belief update operators. The main results are:

Proposition 4. *Every rectification operator satisfies **(R3)** and **(R4)**, but none of them satisfies **(R1)**, **(R2)**, **(R5)**, or **(R6)**.*

Proposition 5. *Every rectification operator satisfies **(U2)**, **(U3)**, **(U4)**, **(U7)**, and **(U8)**, but none of them satisfies **(U1)** or **(U5)**. Some rectification operator satisfies **(U6)** but not all of them.*

The main consequence of Proposition 4 and Proposition 5 is that the family of rectification operators and the families of KM revision (resp. KM update) operators are disjoint. Accordingly, our results clearly show that, from an axiomatic point of view, rectification is a change operation that is clearly distinct from rational revision or update.

5 Some Computational Issues

Computing rectified classifications. We are first interested in the complexity of determining the class of any input instance $\boldsymbol{x}$ when classified by the rectified classification circuit $\Sigma \star T$. Formally:

Definition 8 (classifying from a rectified classifier encoding**).** CLASSIFICATION($\star$) *is the following decision problem:*

- **Input:** *Two propositional representations Σ, T s.t. Σ has the XY-classification property, $\boldsymbol{x} \in \boldsymbol{X}$, $\ell \in L_Y$.*
- **Output:** *Does $(\Sigma \star T)(\boldsymbol{x}) \models \ell$ hold?*

When Σ is a CNF formula obtained by applying Tseitin transformation [Tseitin, 1968] to a XY-classification circuit, the class assignment $\Sigma(\boldsymbol{x})$ associated with a given $\boldsymbol{x}$ can be computed in time linear in $|\Sigma|$ (one can compute the value of every variable of $\Sigma \mid \boldsymbol{x}$ using unit propagation since this mimicks the way the values of the outputs of the gates are computed in a XY-classification circuit each time the values of the inputs are provided). Thus, it is interesting to determine whether such a tractability result for classification still holds when $\Sigma \star T$ is considered instead of Σ. Unfortunately this is not the case, *whatever the rectification operator* $\star$:

Proposition 6. *For every rectification operator $\star$,* CLASSIFICATION($\star$) *is both NP-hard and coNP-hard even if Σ is a CNF formula obtained by applying Tseitin transformation to a XY-classification circuit, $|\Sigma|$ is bounded by a constant, and T is a CNF formula.*

Interestingly, classifying from a rectified classifier encoding is computationally easier for some rectification operators, provided that T is represented in a propositional language that offers some specific properties:

Proposition 7. *Let $\boldsymbol{x} \in \boldsymbol{X}$ be an instance to be classified by a classifier $\boldsymbol{C}$ encoded by Σ, $\boldsymbol{y} \in \boldsymbol{Y}$ be the corresponding prediction (i.e., $\boldsymbol{y} = \boldsymbol{C}(\boldsymbol{x}) = \Sigma(\boldsymbol{x})$), and T be be a propositional representation from a language that supports in polynomial time* **CD**, **FO**, *and* **CO**. *Then:*

- *$(\Sigma \star_D T)(\boldsymbol{x})$ can be computed in time polynomial in $n + m + |T|$.*
- *If s is any selection function that runs in time polynomial in the size of its input, then $(\Sigma \star^s T)(\boldsymbol{x})$ can be computed in time polynomial in $n + m + |T|$.*

What makes the last proposition useful is the existence of succinct languages $\mathcal{L}$ supporting **CD**, **FO**, and **CO**, especially the DNNF language [Darwiche, 2001], and the existence of translators (alias compilers) for turning representations from more standard languages (in particular the CNF one) into DNNF representations.

Measuring how much a classifier complies with a background theory. In order to verify a classifier $\boldsymbol{C}$ via its encoding Σ, it is useful to evaluate the proportion of $\boldsymbol{x} \in \boldsymbol{X}$ for which Σ complies with the background theory T on $\boldsymbol{x}$. If this proportion is high, there is no problem since one expects a high level of compatibility between $\boldsymbol{C}$ and T. Otherwise, it may prove reasonable to learn $\boldsymbol{C}$ again using another training set, or to clean the training set used before a second training round. One step further, it can prove useful to determine such a proportion when the instances under consideration are compatible with a given feature assignment x or $\overline{x}$ with $x \in X$, or more generally with a combination $\boldsymbol{x}'$ of such feature assignments. For instance, when the proportion is lower under assignment x than in the case no feature assignment has been considered, this may reflect incorrect values for x in the training set. To make it more formal, let us consider the following definition:

Definition 9 (knowledge compliance degree)**.** *Given two propositional representations Σ and T such that Σ has the XY-classification property, and $\boldsymbol{x}' \in \boldsymbol{X}'$ with $X' \subseteq X$, the* knowledge compliance degree $kcd(\Sigma, T, \boldsymbol{x}')$ *of Σ with T given $\boldsymbol{x}'$ is given by $kcd(\Sigma, T, \boldsymbol{x}') =$*

$$\frac{\#(\{\boldsymbol{x}'' \in \boldsymbol{X} \setminus \boldsymbol{X}' \mid \Sigma((\boldsymbol{x}', \boldsymbol{x}'')) \models T((\boldsymbol{x}', \boldsymbol{x}''))\})}{2^{|X \setminus X'|}}.$$

Unsurprisingly, computing the knowledge compliance degree of Σ with T given $\boldsymbol{x}'$ is computationally demanding:

Proposition 8. *Computing $kcd(\Sigma, T, \boldsymbol{x}')$ is #P-hard, even under the restriction when T is a monotone 2-CNF formula.*

When additional assumptions on the representations of Σ and T are made, computing $kcd(\Sigma, T, \boldsymbol{x}')$ becomes tractable:

Proposition 9. *If Σ and T belong to a language $\mathcal{L}$ that supports $\wedge\mathbf{BC}$, $\mathbf{CD}$, and $\mathbf{CT}$ and T is such that $Var(T) \subseteq X \cup Y$, then for any $\boldsymbol{x}'$, $kcd(\Sigma, T, \boldsymbol{x}')$ can be computed in time polynomial in the input size.*

Again, what makes the last proposition valuable is the existence of succinct representation languages $\mathcal{L}$ supporting $\wedge\mathbf{BC}$, $\mathbf{CD}$, and $\mathbf{CT}$, especially the languages of (structured) d-DNNF respecting a fixed vtree [Pipatsrisawat and Darwiche, 2008; Pipatsrisawat and Darwiche, 2010] and its subsets, SDD [Darwiche, 2011] and OBDD [Bryant, 1986], as well as the existence of compilers targeting those languages.

6 Other Related Work

Theory revision is a research question that has stimulated much effort in the ML community since the 90's (see e.g., [Wrobel, 1994]). In a nutshell, theory revision is the problem of correcting a given, roughly correct, theory Σ. Typically, Σ represents pieces of knowledge, provided by an expert, and it comes with a set of labelled instances $\boldsymbol{x}$ (examples and counter-examples of a concept). Most of the time, Σ is a logical representation, based on atoms describing features and concepts (or classes), and $\boldsymbol{x}$ is classified by Σ as an element of class C (respresented by an atom) whenever C can be deduced from Σ and $\boldsymbol{x}$ [Ourston and Mooney, 1990]. When an instance is not recognized as it should be, Σ must be minimally modified so as to classify all instances correctly, and this is the purpose of theory revision. A related issue is to minimize the number of syntactic revision operations (such as the addition or deletion of literals) needed to obtain the target theory from Σ [Goldsmith *et al.*, 2002].

Rectification and theory revision are connected by some common goal (correcting misclassifications). Furthermore, like rectification, theory revision does not amount to belief revision: revising a theory by an example using a belief revision operator would enforce the instance to hold in the revised theory, which is not the same as ensuring the revised theory to classify the instance in the right way. However, rectification and theory revision are based on different inputs (in the rectification setting, change formulae do not reduce to instances but can be complex representations linking features and classes) and assumptions (in theory revision, the theory considered at start is not required to classify every instance, even if it is in an incorrect way, and this property is not expected to be maintained) and for those reasons, rectification operators differ from theory revision operators (especially, the former ones are typically syntax-dependent while the latter ones are not).

7 Conclusion

We have presented a new belief change operation, called rectification, that is specific to multi-label classifier encodings Σ. Such an operation aims to minimally modify such an encoding Σ so as to make it to comply with some background theory T, that is considered more reliable than Σ. Several notions of compliance have been presented. Focusing mainly on fact-compliance, we have characterized the family of rectification operators $\star$ from an axiomatic perspective and have exhibited some operators from this family. We have identified the standard revision and/or update postulates that every rectification operator satisfies and those it does not. We have also identified complexity bounds for the problems of computing rectified classifications and of measuring the knowledge compliance of a given Σ with a given T. Though both problems are computationally hard, we have found tractable restrictions of them based on the DNNF language and its subsets. Accordingly, those languages appear as valuable for representing classifier encodings given that they are already known as useful for ensuring the tractability of a number of XAI queries [Audemard *et al.*, 2020].

This work is a step towards leveraging some background knowledge T within the classification circuit Σ associated with a classifier $\boldsymbol{C}$, taking advantage of the fact that T is more reliable than the data from which $\boldsymbol{C}$ has been learned. Being a first step, the present work calls for a number of perspectives. One of them consists in deriving representation theorems for the rectification operators, giving constructive characterization results for their family. We plan also to study in more depth the family of operators ensuring knowledge-compliance, i.e., operators satisfying **(RE2')** and **(RE3')** instead of **(RE2)** and **(RE3)**. Finally, another important perspective for further research concerns the representation side: the rectified representations $\Sigma \star_D T$ and $\Sigma \star^s T$, as given by Definition 7, are of size exponential in the number n of features. This makes their computation impractical most of the time. In order to address this issue, more succinct representations for $\Sigma \star_D T$ and $\Sigma \star^s T$ must be looked for.

Acknowledgements

Many thanks to the anonymous reviewers for their comments and suggestions. This work has benefited from the support of the AI Chair EXPEKCTATION of the French National Research Agency (ANR). It was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

[Audemard *et al.*, 2020] G. Audemard, F. Koriche, and P. Marquis. On tractable XAI queries based on compiled representations. In *Proc. of KR'20*, pages 838–849, 2020.

[Besold *et al.*, 2017] T. R. Besold, A. S. d'Avila Garcez, S. Bader, H. Bowman, P. M. Domingos, P. Hitzler, K. Kühnberger, L. C. Lamb, D. Lowd, P. Machado Vieira Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902, 2017.

[Bryant, 1986] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–692, 1986.

[Chen *et al.*, 2020] Y. Chen, A. Choi, and A. Darwiche. Supervised learning with background knowledge. In *Proc. of PGM'20*, 2020.

[Creignou *et al.*, 2014] N. Creignou, O. Papini, R. Pichler, and S. Woltran. Belief revision within fragments of propositional logic. *J. Comput. Syst. Sci.*, 80(2):427–449, 2014.

[Dalal, 1988] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of AAAI'88*, pages 475–479, 1988.

[Darwiche and Hirth, 2020] A. Darwiche and A. Hirth. On the reasons behind decisions. In *Proc. of ECAI'20*, pages 712–720, 2020.

[Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[Darwiche, 2001] A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[Darwiche, 2011] A. Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *Proc. of IJCAI'11*, pages 819–826, 2011.

[d'Avila Garcez *et al.*, 2019] A. S. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6(4):611–632, 2019.

[Donadello *et al.*, 2017] I. Donadello, L. Serafini, and A. S. d'Avila Garcez. Logic tensor networks for semantic image interpretation. In *Proc. of IJCAI'17*, pages 1596–1602, 2017.

[Goldsmith *et al.*, 2002] J. Goldsmith, R. H. Sloan, and G. Turán. Theory revision with queries: DNF formulas. *Machine Learning*, 47(2-3):257–295, 2002.

[Hu *et al.*, 2016] Z. Hu, X. Ma, Z. Liu, E. H. Hovy, and E. P. Xing. Harnessing deep neural networks with logic rules. In *Proc. of ACL'16*, 2016.

[Katsuno and Mendelzon, 1991a] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proc. of KR'91*, pages 387–394, 1991.

[Katsuno and Mendelzon, 1991b] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

[Lang and Marquis, 2008] J. Lang and P. Marquis. On propositional definability. *Artificial Intelligence*, 172(8-9):991–1017, 2008.

[Narodytska *et al.*, 2018] N. Narodytska, S. Prasad Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh. Verifying properties of binarized deep neural networks. In *Proc. of AAAI'18*, pages 6615–6624, 2018.

[Ourston and Mooney, 1990] D. Ourston and R. J. Mooney. Changing the rules: A comprehensive approach to theory refinement. In *Proc. of AAAI'90*, pages 815–820, 1990.

[Papadimitriou, 1994] Ch. H. Papadimitriou. *Computational complexity*. Addison–Wesley, 1994.

[Pipatsrisawat and Darwiche, 2008] K. Pipatsrisawat and A. Darwiche. New compilation languages based on structured decomposability. In *Proc. of AAAI'08*, pages 517–522, 2008.

[Pipatsrisawat and Darwiche, 2010] K. Pipatsrisawat and A. Darwiche. Top-down algorithms for constructing structured DNNF: Theoretical and practical implications. In *Proc. of ECAI'10*, pages 3–8, 2010.

[Raedt *et al.*, 2016] L. De Raedt, K. Kersting, S. Natarajan, and D. Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Morgan & Claypool Publishers, 2016.

[Raedt *et al.*, 2020] L. De Raedt, S. Dumancic, R. Manhaeve, and G. Marra. From statistical relational to neuro-symbolic artificial intelligence. In *Proc. of IJCAI'20*, pages 4943–4950, 2020.

[Russell, 2015] S. J. Russell. Unifying logic and probability. *Communications of the ACM*, 58(7):88–97, 2015.

[Shi *et al.*, 2020] W. Shi, A. Shih, A. Darwiche, and A. Choi. On tractable representations of binary neural networks. In *Proc. of KR'20*, pages 882–892, 2020.

[Shih *et al.*, 2019] A. Shih, A. Choi, and A. Darwiche. Compiling Bayesian networks into decision graphs. In *Proc. of AAAI'19*, pages 7966–7974, 2019.

[Tseitin, 1968] G.S. Tseitin. *On the complexity of derivation in propositional calculus*, chapter Structures in Constructive Mathematics and Mathematical Logic, pages 115–125. Steklov Mathematical Institute, 1968.

[Wrobel, 1994] S. Wrobel. *Concept Formation and Knowledge Revision*. Springer, 1994.

[Xie *et al.*, 2019] Y. Xie, Z. Xu, K. S. Meel, M. S. Kankanhalli, and H. Soh. Embedding symbolic knowledge into deep networks. In *Proc. of NeurIPS'19*, pages 4235–4245, 2019.

[Xu *et al.*, 2018] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proc. of ICML'18*, volume 80, pages 5498–5507, 2018.