Fuzzy OWL-Boost: Learning Fuzzy Concept Inclusions via Real-Valued Boosting

Franco Alberto Cardillo
ILC - CNR
Pisa, Italy

Umberto Straccia
ISTI - CNR
Pisa, Italy

August 13, 2020

Abstract

OWL ontologies are nowadays a quite popular way to describe structured knowledge in terms of classes, relations among classes and class instances.

In this paper, given a target class T of an OWL ontology, we address the problem of learning fuzzy concept inclusion axioms that describe sufficient conditions for being an individual instance of T. To do so, we present Fuzzy OWL-Boost that relies on the $\mathbb{R}$eal AdaBoost boosting algorithm adapted to the (fuzzy) OWL case. We illustrate its effectiveness by means of an experimentation. An interesting feature is that the learned rules can be represented directly into Fuzzy OWL 2. As a consequence, any Fuzzy OWL 2 reasoner can then be used to automatically determine/classify (and to which degree) whether an individual belongs to the target class T.

1 Introduction

OWL 2 ontologies [88] are nowadays a popular means to represent *structured* knowledge and its formal semantics is based on *Description Logics* (DLs) [4]. The basic ingredients of DLs are concept descriptions (in First-Order Logic terminology, unary predicates), inheritance relationships among them and instances of them.

Although an important amount of work has been carried about DLs, the application of machine learning techniques to OWL 2 ontologies, *viz.* DL ontologies, is relatively less addressed compared to the *Inductive Logic Programming* (ILP) setting (see *e.g.* [91, 92] for more insights on ILP). We refer the reader to [71, 93] for an overview and to Section 2

In this work, we focus on the problem of automatically learning fuzzy concept inclusion axioms from OWL 2 ontologies. More specifically, given a target class T of an OWL ontology, we address the problem of learning fuzzy $\mathcal{EL}(\mathbf{D})$ [3] concept inclusion axioms that describe sufficient conditions for being an individual instance of T.

Example 1.1 (Running example [68, 70, 114]) *Consider an ontology that describes the meaningful entities of a city.* [1] *Now, one may fix a city, say Pisa, extract the properties of the hotels from Web sites, such as location, price, etc., and the hotel judgements of the users,* e.g., *from Trip Advisor.* [2] *Now, using the terminology of the ontology, one may ask about what characterises* good *hotels in Pisa (our target class T) according to the user feedback. Then one may learn from the user feedback that, for instance, 'An expensive Bed and Breakfast is a good hotel' (see also Section 5 later on).* □

The objective is essentially the same as in *e.g.* [70, 114] except that now we propose to rely on the $\mathbb{R}$eal AdaBoost [86] boosting algorithm to be adapted to the (fuzzy) OWL case. Of course, like in [68, 114], we continue to support so-called *fuzzy concept descriptions* and *fuzzy concrete domains* [76, 112, 113] such as 'an *expensive* Bed and Breakfast is a good hotel'. Here, the concept *expensive* is a so-called fuzzy concept [123], *i.e.* a concept for which the belonging of an individual to the class is not necessarily a binary yes/no question, but rather a matter of degree in $[0, 1]$. For instance, in our example, the degree of expensiveness of a hotel may depend on the price of the hotel: the higher the price the more expensive is

[1] For instance, http://donghee.info/research/SHSS/ObjectiveConceptsOntology(OCO).html
[2] http://www.tripadvisor.com

the hotel. Here, the range of the 'attribute' *hotel price* becomes a so-called fuzzy concrete domain [113] allowing to specify fuzzy labels such as 'high/moderate/low price'.

We recall that (discrete) AdaBoost [46, 108, 47] uses weak hypotheses with outputs restricted to the discrete set of classes that it combines via leveraging weights in a linear vote. On the other hand $\mathbb{R}$eal AdaBoost [86] is a generalisation of it as real-valued weak hypotheses are admitted (see [86] for a comparison to approaches to real-valued AdaBoost).

Besides the fact that (to the best of our knowledge) the use of both (discrete) AdaBoost (with the notable exception of [44]) and its generalisation to real-valued weak hypotheses in the context OWL 2 ontologies is essentially unexplored, the main features of our algorithm, called Fuzzy OWL-Boost, are the following:

- it generates a set of fuzzy fuzzy $\mathcal{EL}(\mathbf{D})$ inclusion axioms [14], which are the weak hypothesis, possibly including fuzzy concepts and fuzzy concrete domains [76, 112, 113], where each axiom has a leveraging weight;
- the fuzzy concept inclusion axioms are then linearly combined into a new fuzzy concept inclusion axiom describing sufficient conditions for being an individual instance of the target class T;
- all generated fuzzy concept inclusion axioms can then be directly encoded as *Fuzzy OWL 2* axioms [11, 12].[3] As a consequence, a Fuzzy OWL 2 reasoner, such as *fuzzyDL* [10, 13], can then be used to automatically determine (and to which degree) whether an individual belongs to the target class T.

Let us remark that we rely on real-valued AdaBoost as the weak hypotheses Fuzzy OWL-Boost generates are indeed fuzzy concept inclusion axioms and, thus, the degree to which an instance satisfies them is a real-valued degree of truth in $[0, 1]$.

In the following, we proceed as follows. In Section 2 we compare our work with closely related work appeared so far. For completeness, we refer to A in which we provide a much more extensive list of references related to OWL rule learning, though less related to our setting. In Section 3, for the sake of completeness, we recap the salient notions we will rely on in this paper. Then, in Section 4 we will present our algorithm Fuzzy OWL-Boost, which then is evaluated for its effectiveness in Section 5. Section 6 concludes and points to some topics of further research.

2 Related Work

Concepts inclusion axioms learning in DLs stems from statistical relational learning, where classification rules are (possibly weighted) Horn clause theories from examples (see *e.g.* [91, 92]) and various methods have been proposed in the DL context so far (see *e.g.* [71, 93]). The general idea consists of the exploration of the search space of potential concept descriptions that cover the available training examples using so-called refinement operators (see, *e.g.* [5, 59, 62]). The goal is then to learn a concept description of the underlying DL language covering (possibly) all provided positive examples and (possibly) not covering any of the provided negative examples. The fuzzy case (see [67, 70, 114]) is a natural extension in which one relies on fuzzy DLs [9, 113] and fuzzy ILP (see *e.g.* [109]) instead.

Closely related to our work are [44, 67, 70, 114]. The works [67, 70], which stem essentially from [68, 69, 72, 73, 74, 75], propose fuzzy Foil-like algorithms and are inspired by fuzzy ILP variants such as [29, 109, 111],[4] while here we rely on a real-valued variant of AdaBoost. Let us note that [67, 73] consider the weaker hypothesis representation language DL-Lite [2], while here we rely on fuzzy $\mathcal{EL}(\mathbf{D})$ as in [68, 69, 72, 74, 75, 70]. Fuzzy $\mathcal{EL}(\mathbf{D})$ has also been considered in [114], which however differs from [67, 70] by the fact that a (fuzzy) probabilistic ensemble evaluation of the fuzzy concept description candidates has been considered. [5] Discrete boosting has been considered in [44], which also shows how to derive a weak learner —(called wDLF) from conventional learners using some sort of random downward refinement operator covering at least a positive example and yielding a minimal score fixed with a threshold. Besides that we deal here with fuzziness in the hypothesis language and a real-valued variant of AdaBoost, the weak learner we propose here differentiates from the previous one by using a kind of gradient descent like algorithm to

[3] As Fuzzy OWL 2 supports the linear combination of weighted concepts.

[4] See, *e.g.* [19], for an overview on fuzzy rule learning mehtods.

[5] Also, as far as we were able to figure out, concrete datatypes were not addressed in the evaluation.

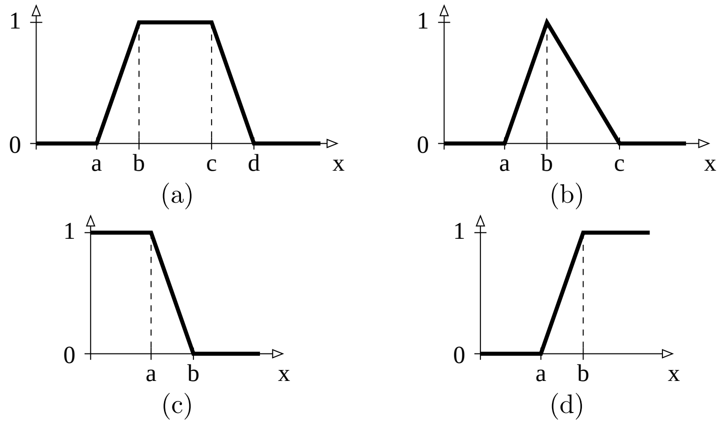

Figure 1: (a) Trapezoidal function $trz(a, b, c, d)$, (b) triangular function $tri(a, b, c)$, (c) left shoulder function $ls(a, b)$, and (d) right shoulder function $rs(a, b)$.

search for the best alternative. Notably, this also deviates from 'fuzzy' rule learning AdaBoost variants, such as [28, 87, 90, 107, 122] in which the weak learner is required to generate the whole rules search space beforehand the selection of the best current alternative. Such an approach is essentially unfeasible in the OWL case due to the size of the search space.

Eventually, [53] can learn fuzzy OWL DL concept equivalence axioms from FuzzyOWL 2 ontologies, by interfacing with the *fuzzyDL* reasoner [13]. The candidate concept expressions are provided by the underlying DL-Learner [57, 15, 16] system. However, it has been tested only on a toy ontology so far. Last, but not least, let us mention [55], which is based on an ad-hoc translation of fuzzy Łukasiewicz $\mathcal{ALC}$ DL constructs into fuzzy *Logic Programming* (fuzzy LP) and then uses a conventional ILP method to learn rules. Unfortunately, the method is not sound as it has been shown that the mapping from fuzzy DLs to LP is incomplete [83] and entailment in Łukasiewicz $\mathcal{ALC}$ is undecidable [17].

While it is not our aim here to provide an extensive overview about learning w.r.t. ontologies literature, nevertheless we refer the interested reader to A for an extensive list of references, which may be the subject of a survey paper instead.

3 Background

For the sake of self completeness, we first introduce the main notions related to *(Mathematical) Fuzzy Logics* and *Fuzzy Description Logics* we will use in this work (see [113] for a more extensive introduction to both).

3.1 Mathematical Fuzzy Logic

Fuzzy Logic is the logic of *fuzzy sets* [123]. A *fuzzy set* A over a countable crisp set X is a function $A \colon X \to [0, 1]$, called *fuzzy membership* function of A. A crisp set A is characterised by a membership function $A \colon X \to \{0, 1\}$ instead. The 'standard' fuzzy set operations conform to $(A \cap B)(x) = \min(A(x), B(x))$, $(A \cup B)(x) = \max(A(x), B(x))$ and $\bar{A}(x) = 1 - A(x)$ ($\bar{A}$ is the set complement of A), the *cardinality* of a fuzzy set is often defined as $|A| = \sum_{x \in X} A(x)$, while the *inclusion degree* between A and B is defined typically as $deg(A, B) = \frac{|A \cap B|}{|A|}$.

The trapezoidal (Fig. 1 (a)), the triangular (Fig. 1 (b)), the L-function (left-shoulder function, Fig. 1 (c)), and the R-function (right-shoulder function, Fig. 1 (d)) are frequently used to specify membership functions of fuzzy sets.

Although fuzzy sets have a greater expressive power than classical crisp sets, their usefulness depends critically on the capability to construct appropriate membership functions for various given concepts in different contexts. We refer the interested reader to, *e.g.*, [54]. One easy and typically satisfactory method to define the membership functions is to uniformly partition the range of, *e.g.* salary values (bounded by a minimum and maximum value), into 5 or 7 fuzzy sets using triangular (or trapezoidal) functions

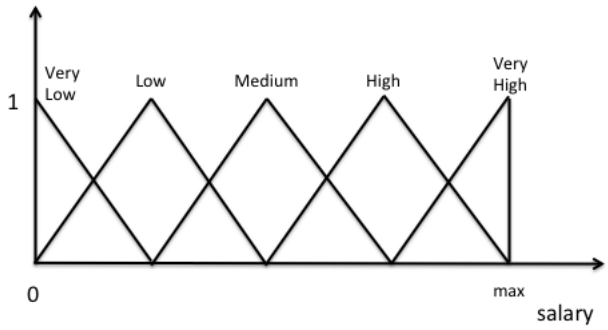

Figure 2: Uniform fuzzy sets over salaries.

Table 1: Combination functions for fuzzy logics.

	Łukasiewicz	Gödel	Product	standard
$d_1 \otimes d_2$	$\max(d_1 + d_2 - 1, 0)$	$\min(d_1, d_2)$	$d_1 \cdot d_2$	$\min(d_1, d_2)$
$d_1 \oplus d_2$	$\min(d_1 + d_2, 1)$	$\max(d_1, d_2)$	$d_1 + d_2 - d_1 \cdot d_2$	$\max(d_1, d_2)$
$d_1 \Rightarrow d_2$	$\min(1 - d_1 + d_2, 1)$	$\begin{cases} 1 & \text{if } d_1 \leq d_2 \\ \beta & \text{otherwise} \end{cases}$	$\min(1, d_2/d_1)$	$\max(1 - d_1, d_2)$
$\ominus d$	$1 - d$	$\begin{cases} 1 & \text{if } d = 0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } d = 0 \\ 0 & \text{otherwise} \end{cases}$	$1 - d$

(see Figure 2). Another popular approach may consist in using the so-called *C-means* fuzzy clustering algorithm (see, *e.g.* [7]) with three or five clusters, where the fuzzy membership functions are triangular functions built around the centroids of the clusters (see also [51]).

In *Mathematical Fuzzy Logic* [48], the convention prescribing that a formula is either true or false (w.r.t. an interpretation $\mathcal{I}$) is changed and is a matter of degree measured on an ordered scale that is no longer $\{0, 1\}$, but typically $[0, 1]$. This degree is called *degree of truth* of the formula ϕ in the interpretation $\mathcal{I}$. Here, *fuzzy formulae* have the form $\langle \phi, d \rangle$, where $d \in (0, 1]$ and ϕ is a First-Order Logic (FOL) formula, encoding that the degree of truth of ϕ is *greater than or equal to d*. So, for instance, $\langle Cheap(HotelVerdi), 0.8 \rangle$ states that 'Hotel Verdi is cheap' is true to degree greater or equal 0.8. From a semantics point of view, a *fuzzy interpretation* $\mathcal{I}$ maps each atomic formula p_i into $[0, 1]$ and is then extended inductively to all FOL formulae as follows:

$$
\begin{array}{rcl}
\mathcal{I}(\phi \land \psi) & = & \mathcal{I}(\phi) \otimes \mathcal{I}(\psi) \ \ , \ \ \mathcal{I}(\phi \lor \psi) \ = \ \mathcal{I}(\phi) \oplus \mathcal{I}(\psi) \\
\mathcal{I}(\phi \to \psi) & = & \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi) \ \ , \ \ \mathcal{I}(\neg \phi) \ = \ \ominus \mathcal{I}(\phi) \\
\mathcal{I}(\exists x.\phi(x)) & = & \sup_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y)) \ \ , \ \ \mathcal{I}(\forall x.\phi(x)) \ = \ \inf_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y)) \ ,
\end{array}
$$

where $\Delta^{\mathcal{I}}$ is the domain of $\mathcal{I}$, and $\otimes$, $\oplus$, $\Rightarrow$, and $\ominus$ are so-called *t-norms*, *t-conorms*, *implication functions*, and *negation functions*, respectively, which extend the Boolean conjunction, disjunction, implication, and negation, respectively, to the fuzzy case.

One usually distinguishes three different logics, namely Łukasiewicz, Gödel, and Product logics [48], [6] whose truth combination functions are reported in Table 1.

Note that the operators for 'standard' fuzzy logic, namely $d_1 \otimes d_2 = \min(d_1, d_2)$, $d_1 \oplus d_2 = \max(d_1, d_2)$, $\ominus d = 1 - d$ and $d_1 \Rightarrow d_2 = \max(1 - d_1, d_2)$, can be expressed in Łukasiewicz logic. More precisely, $\min(d_1, d_2) = d_1 \otimes_l (d_1 \Rightarrow_l d_2), \max(d_1, d_2) = 1 - \min(1 - d_1, 1 - d_2)$. Furthermore, the implication $d_1 \Rightarrow_{kd} d_2 = \max(1 - d_1, d_2)$ is called *Kleene-Dienes implication* (denoted $\Rightarrow_{kd}$), while *Zadeh implication* (denoted $\Rightarrow_z$) is the implication $d_1 \Rightarrow_z d_2 = 1$ if $d_1 \leq d_2$; 0 otherwise.

An *r-implication* is an implication function obtained as the *residuum* of a continuous t-norm $\otimes$, [7]

[6]Notably, a theorem states that any other continuous t-norm can be obtained as a combination of them.

[7]Note that Łukasiewicz, Gödel and Product implications are r-implications, while Kleene-Dienes implication is not.

i.e. $d_1 \Rightarrow d_2 = \max\{d_3 \mid d_1 \otimes d_3 \leq d_2\}$. Note also, that given an r-implication $\Rightarrow_r$, we may also define its related negation $\ominus_r d$ by means of $d \Rightarrow_r 0$ for every $d \in [0,1]$.

The notions of satisfiability and logical consequence are defined in the standard way, where a fuzzy interpretation $\mathcal{I}$ *satisfies* a fuzzy formula $\langle \phi, d\rangle$, or $\mathcal{I}$ is a *model* of $\langle \phi, d\rangle$, denoted as $\mathcal{I} \models \langle \phi, d\rangle$, iff $\mathcal{I}(\phi) \geq d$. Notably, from $\langle \phi, d_1\rangle$ and $\langle \phi \to \psi, d_2\rangle$ one may conclude (if $\to$ is an r-implication) $\langle \psi, d_1 \otimes d_2\rangle$ (this inference is called *fuzzy modus ponens*).

3.2 Fuzzy Description Logics basics

We recap here the fuzzy DL $\mathcal{ALCW}(\mathbf{D})$, which extends the well-known fuzzy DL $\mathcal{ALC}(\mathbf{D})$ [112] with the *weighted concept* construct (indicated with the letter $\mathcal{W}$) [12, 113]. $\mathcal{ALCW}(\mathbf{D})$ is expressive enough to capture the main ingredients of fuzzy DLs we are going to consider here. Note that fuzzy DLs and fuzzy OWL 2 in particular, cover many more language constructs than we use here (see, *e.g.* [9, 12, 113]).

We start with the notion of *fuzzy concrete domain*, that is a tuple $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ with datatype domain $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each data value an element of $\Delta^{\mathbf{D}}$, and to every 1-ary datatype predicate $\mathbf{d}$ a 1-ary fuzzy relation over $\Delta^{\mathbf{D}}$. Therefore, $\cdot^{\mathbf{D}}$ maps indeed each datatype predicate into a function from $\Delta^{\mathbf{D}}$ to $[0,1]$. Typical datatypes predicates $\mathbf{d}$ are characterized by the well known membership functions (see also Fig. 1)

$$
\begin{array}{rcl}
\mathbf{d} & \to & ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \\
 & & \mid \ \geq_v \ \mid \ \leq_v \ \mid \ =_v \ ,
\end{array}
$$

where *e.g.* $ls(a,b)$ is the left-shoulder membership function and $\geq_v$ corresponds to the crisp set of data values that are greater than or equal to the value v.

Now, consider pairwise disjoint alphabets $\mathbf{I}$, $\mathbf{A}$ and $\mathbf{R}$, where $\mathbf{I}$ is the set of *individuals*, $\mathbf{A}$ is the set of *concept names* (also called *atomic concepts*) and $\mathbf{R}$ is the set of *role names*. Each role is either an *object property* or a *datatype property*. The set of *concepts* are built from concept names A using connectives and quantification constructs over object properties R and datatype properties S, as described by the following syntactic rule ($n \geq 1, \alpha_i \in (0,1], \sum_i \alpha_i \leq 1$):

$$
\begin{array}{rcl}
C & \to & \top \mid \ \bot \ \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid C_1 \to C_2 \mid \\
 & & \exists R.C \mid \forall R.C \mid \exists S.\mathbf{d} \mid \forall S.\mathbf{d} \mid \\
 & & \alpha_1 \cdot C_1 + \ldots \alpha_n \cdot C_n \ .
\end{array}
$$

An *ABox* $\mathcal{A}$ consists of a finite set of assertion axioms. An *assertion* axiom is an expression of the form $\langle a{:}C, d\rangle$ (called *concept assertion*, a is an instance of concept C to degree greater than or equal to d) or of the form $\langle (a_1, a_2){:}R, d\rangle$ (called *role assertion*, (a_1, a_2) is an instance of object property R to degree greater than or equal to d), where a, a_1, a_2 are individual names, C is a concept, R is an object property and $d \in (0,1]$ is a truth value. A *Terminological Box* or *TBox* $\mathcal{T}$ is a finite set of *General Concept Inclusion* (GCI) axioms, where a fuzzy GCI is of the form $\langle C_1 \sqsubseteq C_2, d\rangle$ (C_1 is a sub-concept of C_2 to degree greater than or equal to$d$), where C_i is a concept and $d \in (0,1]$. We may omit the truth degree d of an axiom; in this case $d = 1$ is assumed and we call the axiom *crisp*. We also write $C_1 = C_2$ as a macro for the two GCIs $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$. We may also call a fuzzy GCI of the form $\langle C \sqsubseteq A, d\rangle$, where A is a concept name, a *rule* and C its *body*. A *Knowledge Base* (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox. With $\mathsf{I}_{\mathcal{K}}$ we denote the set of individuals occurring in $\mathcal{K}$.

Concerning the semantics, let us fix a fuzzy logic and a fuzzy concrete domain $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$. Now, unlike classical DLs in which an interpretation $\mathcal{I}$ maps *e.g.* a concept C into a set of individuals $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, *i.e.* $\mathcal{I}$ maps C into a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to \{0,1\}$ (either an individual belongs to the extension of C or does not belong to it), in fuzzy DLs, $\mathcal{I}$ maps C into a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and, thus, an individual belongs to the extension of C to some degree in $[0,1]$, *i.e.* $C^{\mathcal{I}}$ is a fuzzy set. Specifically, a *fuzzy interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a nonempty (crisp) set $\Delta^{\mathcal{I}}$ (the *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns: *(i)* to each atomic concept A a function $A^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \to [0,1]$; *(ii)* to each object property R a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$; *(iii)* to each datatype property S a function $S^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}} \to [0,1]$; *(iv)* to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (the so-called *Unique Name Assumption*); and *(v)* to each data value v an element $v^{\mathcal{I}} \in \Delta^{\mathbf{D}}$. Now, a fuzzy interpretation function is extended to concepts as specified below (where $x \in \Delta^{\mathcal{I}}$):

$$\begin{array}{rcl}
\top^{\mathcal{I}}(x) & = & 1 \\
\bot^{\mathcal{I}}(x) & = & 0 \\
(C \sqcap D)^{\mathcal{I}}(x) & = & C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x) \\
(C \sqcup D)^{\mathcal{I}}(x) & = & C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x) \\
(\neg C)^{\mathcal{I}}(x) & = & \ominus C^{\mathcal{I}}(x) \\
(C \rightarrow D)^{\mathcal{I}}(x) & = & C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \\
(\forall R.C)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\} \\
(\exists R.C)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\} \\
(\forall S.\mathbf{d})^{\mathcal{I}}(x) & = & \inf_{y \in \Delta^{\mathbf{D}}} \{S^{\mathcal{I}}(x,y) \Rightarrow \mathbf{d}^{\mathbf{D}}(y)\} \\
(\exists S.\mathbf{d})^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathbf{D}}} \{S^{\mathcal{I}}(x,y) \otimes \mathbf{d}^{\mathbf{D}}(y)\} \\
(\alpha_1 \cdot C_1 + \ldots \alpha_n \cdot C_n)^{\mathcal{I}}(x) & = & \sum_i \alpha_i \cdot C_i{}^{\mathcal{I}}(x) \ .
\end{array}$$

The *satisfiability of axioms* is then defined by the following conditions: *(i)* $\mathcal{I}$ satisfies an axiom $\langle a{:}C, d\rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq d$; *(ii)* $\mathcal{I}$ satisfies an axiom $\langle (a,b){:}R, d\rangle$ if $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq d$; *(iii)* $\mathcal{I}$ satisfies an axiom $\langle C \sqsubseteq D, d\rangle$ if $(C \sqsubseteq D)^{\mathcal{I}} \geq d$ with [8] $(C \sqsubseteq D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)\}$. $\mathcal{I}$ is a *model* of $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}\rangle$ iff $\mathcal{I}$ satisfies each axiom in $\mathcal{K}$. If $\mathcal{K}$ has a model we say that $\mathcal{K}$ is *satisfiable* (or *consistent*). We say that $\mathcal{K}$ *entails* axiom τ, denoted $\mathcal{K} \models \tau$, if any model of $\mathcal{K}$ satisfies τ. The *best entailment degree* of τ of the form $C \sqsubseteq D$, $a{:}C$ or $(a,b){:}R$, denoted $bed(\mathcal{K}, \tau)$, is defined as

$$bed(\mathcal{K}, \tau) = \sup\{d \mid \mathcal{K} \models \langle \tau, d\rangle\} \ .$$

Remark 1 *Please note that $bed(\mathcal{K}, a{:}C) = 1$ (i.e. $\mathcal{K} \models a{:}C$) implies $bed(\mathcal{K}, a{:}\neg C) = 0$, and similarly, $bed(\mathcal{K}, a{:}\neg C) = 1$ (i.e. $\mathcal{K} \models a{:}\neg C$) implies $bed(\mathcal{K}, a{:}C) = 0$. However, in both cases the other way around does not hold. Furthermore, we may well have that both $bed(\mathcal{K}, a{:}C) = d_1 > 0$ and $bed(\mathcal{K}, a{:}\neg C) = d_2 > 0$ hold.*

Eventually, consider concept C, a GCI $C \sqsubseteq D$, a KB $\mathcal{K}$, a set of individuals I and a (weight) distribution $\mathbf{w}$ over I. Then the *cardinality* of C w.r.t. $\mathcal{K}$ and I, denoted $|C|^{\mathsf{I}}_{\mathcal{K}}$, is defined as

$$|C|^{\mathsf{I}}_{\mathcal{K}} = \sum_{a \in \mathsf{I}} bed(\mathcal{K}, a{:}C) \ , \tag{1}$$

while the *weighted cardinality* C w.r.t. $\mathcal{K}$, $\mathbf{w}$ and I , denoted $|C|^{\mathbf{w},\mathsf{I}}_{\mathcal{K}}$, is defined as

$$|C|^{\mathbf{w},\mathsf{I}}_{\mathcal{K}} = \sum_{a \in \mathsf{I}} w_a \cdot bed(\mathcal{K}, a{:}C) \ . \tag{2}$$

The *crisp cardinality* (denoted $\lceil C\rceil^{\mathsf{I}}_{\mathcal{K}}$) and *crisp weighted cardinality* (denoted $\lceil C\rceil^{\mathbf{w},\mathsf{I}}_{\mathcal{K}}$) are defined similarly by replacing in Eq. 1 and 2 the term $bed(\mathcal{K}, a{:}C)$ with $\lceil bed(\mathcal{K}, a{:}C)\rceil$.

Furthermore, the *confidence degree* (also called *inclusion degree*) of $C \sqsubseteq D$ w.r.t. $\mathcal{K}$ and I, denoted $cf(C \sqsubseteq D, \mathsf{I})$, is defined as

$$cf(C \sqsubseteq D, \mathsf{I}) = \frac{|C \sqcap D|^{\mathsf{I}}_{\mathcal{K}}}{|C|^{\mathsf{I}}_{\mathcal{K}}} \ . \tag{3}$$

Similarly, the *weighted confidence degree* (also called *weighted inclusion degree*) of $C \sqsubseteq D$ w.r.t. $\mathcal{K}$, $\mathbf{w}$ and I, denoted $cf(C \sqsubseteq D, \mathbf{w}, \mathsf{I})$, is defined as

$$cf(C \sqsubseteq D, \mathbf{w}, \mathsf{I}) = \frac{|C \sqcap D|^{\mathbf{w},\mathsf{I}}_{\mathcal{K}}}{|C|^{\mathbf{w},\mathsf{I}}_{\mathcal{K}}} \ . \tag{4}$$

[8] However, note that under standard logic $\sqsubseteq$ is interpreted as $\Rightarrow_z$ and not as $\Rightarrow_{kd}$.

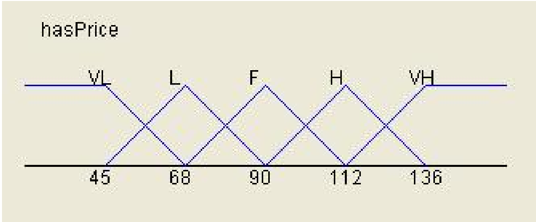

Figure 3: Fuzzy sets derived from the datatype property `hasPrice`.

Example 3.1 (Example 1.1 cont.) *Let us consider the following axiom*

$$\langle \exists hasPrice.High \sqsubseteq GoodHotel, 0.569\rangle \ ,$$

where hasPrice is a datatype property whose values are measured in euros and the price concrete domain has been automatically fuzzified as illustrated in Figure 3. Now, it can be verified that for hotel verdi, whose room price is 105 *euro,* i.e. *we have the assertion* $verdi{:}\exists hasPrice.=_{105}$ *in the KB, we infer under Product logic that* [9]

$$\mathcal{K} \models \langle verdi{:}GoodHotel, 0.18\rangle \ .$$

□

4 Learning Fuzzy Concept Inclusions via Real-Valued Boosting

To start with, we introduce our learning problem.

4.1 The Learning Problem

In general terms, the learning problem we are going to address is stated as follows:
Given:

- a satisfiable KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ and its individuals $\mathsf{I}_{\mathcal{K}}$;
- a *target concept name* T with an associated unknown classification function $f_T \colon \mathsf{I}_{\mathcal{K}} \to \{1, 0\}$, where for each $a \in \mathsf{I}_{\mathcal{K}}$, the possible values (*labels*) correspond, respectively, to $\mathcal{K} \models a{:}T$ (a is a *positive* example of T) and $\mathcal{K} \not\models a{:}T$ (a is a *non-positive* example of T);
- a *hypothesis space* of classifiers $\mathcal{H} = \{h \colon \mathsf{I}_{\mathcal{K}} \to [0, 1]\}$;
- a *training set* $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ (the positive and non-positive examples of T, respectively) of individual-label pairs:

$$\begin{aligned} \mathcal{E}^+ &= \{(a, 1) \mid a \in \mathsf{I}_{\mathcal{K}}, f_T(a) = 1\} \\ \mathcal{E}^- &= \{(a, 0) \mid a \in \mathsf{I}_{\mathcal{K}}, f_T(a) = 0\} \ . \end{aligned}$$

  With $\mathsf{I}_{\mathcal{E}}$ we denote the set of individuals occurring in $\mathcal{E}$. We assume that for all $a \in \mathsf{I}_{\mathcal{E}}$, $0 = bed(\mathcal{K}, a{:}T) = bed(\mathcal{K}, a{:}\neg T)$, *i.e.* both $\mathcal{K} \not\models \langle a{:}T, d\rangle$ and $\mathcal{K} \not\models \langle a{:}\neg T, d\rangle$ hold for all $d > 0$. [10] We write $\mathcal{E}(a) = 1$ if a is a positive example (*i.e.*, $a \in \mathsf{I}_{\mathcal{E}^+}$), $\mathcal{E}(a) = 0$ if a is a non-positive example (*i.e.*, $a \in \mathsf{I}_{\mathcal{E}^-}$).

Learn: a classifier $\bar{h} \in \mathcal{H}$ that is the result of *Emprical Risk Minimisation* (ERM) on $\mathcal{E}$. That is,

$$\begin{aligned} \bar{h} &= \arg\min_{h \in \mathcal{H}} R(h, \mathcal{E}) \\ &= \mathbf{E}_{\mathcal{E}}[L(h(a), \mathcal{E}(a))] \\ &= \frac{1}{|\mathcal{E}|} \sum_{a \in \mathsf{I}_{\mathcal{E}}} L(h(a), \mathcal{E}(a)) \ , \end{aligned}$$

[9] Using fuzzy modus ponens, $0.18 = 0.318 \cdot 0.569$, where $0.318 = tri(90, 112, 136)(105)$.
[10] Essentially we state that $\mathcal{K}$ does not already know whether a is an instance of T or not.

where L is a *loss function* such that $L(\hat{l}, l)$ measures how different the prediction $\hat{l}$ of a hypothesis is from the true outcome l and $R(h, \mathcal{E})$ is the *risk* associated with hypothesis h over $\mathcal{E}$, defined as the expectation of the loss function over $\mathcal{E}$.

The effectiveness of the learned classifier $\bar{h}$ is then assessed by determining $R(\bar{h}, \mathcal{E}')$ on a a *test set* $\mathcal{E}'$, disjoint from $\mathcal{E}$.

In our setting, we assume that a hypothesis $h \in \mathcal{H}$ is a fuzzy GCI of the form

$$\langle \alpha_1 \cdot C_1 + \ldots \alpha_n \cdot C_n \sqsubseteq T, d\rangle \tag{5}$$

where each C_i is a so-called fuzzy $\mathcal{EL}(\mathbf{D})$ concept expression [11] defined according to the following syntax:[12]

$$\begin{array}{lcl} C & \longrightarrow & \top \mid A \mid \exists R.C \mid \exists S.\mathbf{d} \mid C_1 \sqcap C_2 \\ \mathbf{d} & \rightarrow & ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \mid \mathbf{bool} \ . \end{array}$$

For $a \in \mathsf{I}_{\mathcal{K}}$, the *classification prediction value* $h(a)$ of a w.r.t. h, T and $\mathcal{K}$ is defined as (for ease, we omit $\mathcal{K}$ and T)

$$h(a) = bed(\mathcal{K} \cup \{h\}, a{:}T) \ .$$

Note that, as stated above, essentially a hypothesis is a sufficient condition (expressed via the weighted sum of concepts) for being an individual instance of a target concept to some positive degree. So, if $h(a) = 0$ then a is a non-positive instance of T, while if $h(a) > 0$ then a is a positive instance of T to some degree and, thus, we distinguish between positive and non-positive instances of T only. Furthermore, let us note that even if $\mathcal{K}$ is a crisp KB, the possible occurrence of fuzzy concrete domains in expressions of the form $\exists S.\mathbf{d}$ in the left-hand side of a hypothesis may imply that $h(a) \notin \{0, 1\}$.

Remark 2 *Note that in* e.g. *[70] a hypothesis is of the form* $\langle C_1 \sqcup \ldots \sqcup C_n \sqsubseteq T, d\rangle$ *instead.* $\square$

Remark 3 *Clearly, the set of hypothesis by this syntax is potentially infinite due,* e.g.*, to conjunction and the nesting of existential restrictions. The set is made finite by imposing further restrictions on the generation process such as the maximal number of conjuncts and the depth of existential nestings allowed.* $\square$

Remark 4 *One may also think of further partition the set $\mathcal{E}^-$ of non-positive examples into a set $\mathcal{E}^\neg$ of* negative *and a set $\mathcal{E}^u$ of* unknown *examples (and use as labelling set $\{-1, 0, 1\}$, respectively, with* 1 *–positive,* 0 *– unknown,* -1 *– negative), as done in many other approaches (see* e.g. *[44]). That is, an individual a is a* negative *example of T if $\mathcal{K} \models a{:}\neg T$, while a is a* unknown *example of T if neither $\mathcal{K} \models a{:}T$ nor $\mathcal{K} \models a{:}\neg T$ hold. In that case, usually we are looking for an exact definition of T,* i.e. *a hypothesis is of the stronger form $T = C$ instead.* [13] *That is, we may well have the case $D \sqsubseteq T$ and $T = C$ with $D \sqsubseteq C$. Which one to choose may depend on the application domain and on the effectiveness of the approach. We do not address this case here.* $\square$

It is easily verified that indeed a hypothesis $\langle \alpha_1 \cdot C_1 + \ldots \alpha_n \cdot C_n \sqsubseteq T, d\rangle$ can be rewritten as a set of rules of the form (with T_i new concept names):

$$\begin{array}{l} \langle C_1 \sqsubseteq T_1, d_1\rangle \\ \quad\vdots \\ \langle C_n \sqsubseteq T_n, d_n\rangle \\ \langle \beta_1 \cdot T_1 + \ldots \beta_n \cdot T_n \sqsubseteq T, d\rangle \end{array} \tag{6}$$

where, as we will see later on, each fuzzy $\mathcal{EL}(\mathbf{D})$ GCI $\langle C_i \sqsubseteq T_i, d_i\rangle$ is a *weak hypothesis* (classifier), while their aggregation is computed via $\mathbb{R}$eal AdaBoost in which each $\beta_i = \alpha_i / d_i$ indicates how much C_i contributes to the classification prediction value.

Remark 5 *Of course, one may also rewrite Eq. 5 directly as (with T_i new concept names)*

$$\begin{array}{l} C_1 \sqsubseteq T_1 \\ \quad\vdots \\ C_n \sqsubseteq T_n \\ \langle \alpha_1 \cdot T_1 + \ldots \alpha_n \cdot T_n \sqsubseteq T, d\rangle \ . \end{array} \tag{7}$$

However, we prefer to rely on Eq. 6 to maintain the confidence degree of each learned rule. $\square$

[11] Note that $\mathcal{EL}$ is a basic ingredient of the OWL profile language OWL EL [89].

[12] **bool** is the concrete domain of boolean values.

[13] We recall that a hypothesis as in Eq. 5 does not allow us to infer negative instances of T, while $T = C$ does.

We conclude with the notions of *consistent*, *non-renduntant*, *sound*, *complete* and *strongly complete* hypothesis h w.r.t. $\mathcal{K}$, which are defined as follows:

Consistency. $\mathcal{K} \cup \{h\}$ is a consistent;

Non-Redundancy. $\mathcal{K} \not\models h$.

Soundness. $\forall a \in \mathsf{I}_{\mathcal{E}^-}, h(a) = 0$.

Completeness. $\forall a \in \mathsf{I}_{\mathcal{E}^+}, h(a) > 0$.

Strong Completeness. $\forall a \in \mathsf{I}_{\mathcal{E}^+}, h(a) = 1$.

We say that a hypothesis h *covers* (*strongly covers*) an example $e \in \mathcal{E}$ iff $bed(\mathcal{K} \cup \{h\}, e) > 0$ ($bed(\mathcal{K} \cup \{h\}, e) = 1$). Therefore, soundness states that a learned hypothesis is not allowed to cover a non-positive example, while the way (strong) completeness is stated guarantees that all positive examples are (strongly) covered.

In general a learned (*induced*) hypothesis h has to be *consistent*, *non-renduntant* and *sound* w.r.t. $\mathcal{K}$, but not necessarily complete, but, of course, these conditions can also be relaxed.

4.2 The Learning Algorithm

We now present our real-valued boosting-based algorithm, which is based on a boosting schema applied a fuzzy GCI learner. Our learning method creates an *ensemble* of classifiers made up of fuzzy $\mathcal{EL}(\mathbf{D})$ concept expressions (see Eq. 5), each of which is provided by a fuzzy *weak learner*, whose predictiveness is required to be better than randomness. Essentially, at each round the weak learner generates a fuzzy $\mathcal{EL}(\mathbf{D})$ candidate GCI of the form $\langle C_i \sqsubseteq T, d_i \rangle$ that determines a change to the distribution of the weights associated with the examples. The weights of misclassified examples get increased so that a better classifier can be produced in the next round, indicating the harder examples to focus on. The weak hypotheses are then eventually combined into a hypothesis (see Eq. 6). We will rely on $\mathbb{R}$eal AdaBoost [85, 86] as boosting algorithm, while we will use a weak learner that is similar to FOIL-$\mathcal{DL}$ [67, 68, 70], both of which need to adapted to our specific setting.

Formally, consider a KB, $\mathcal{K}$, a training set $\mathcal{E}$, a set of individuals I with $\mathsf{I}_{\mathcal{E}} \subseteq \mathsf{I} \subseteq \mathsf{I}_{\mathcal{K}}$, and a weight distribution $\mathbf{w}$ over I. [14] With $\mathbf{u}$ we indicate the uniform distribution over I, *i.e.* $u_a = 1/|\mathsf{I}|$ (with $a \in \mathsf{I}$). Furthermore, consider a weak hypothesis h_i of the form $\langle C_i \sqsubseteq T, d_i \rangle$ returned by the weak learner. Note that for $a \in \mathsf{I}_{\mathcal{K}}$, $bed(\mathcal{K} \cup \{h_i\}, a{:}T) \in [0, 1]$. Next, we transform this value into a value in $[-1, 1]$ as required by $\mathbb{R}$eal AdaBoost. So, let let $t \colon [0, 1] \to [-1, 1]$ be the transformation function

$$t(x) = \begin{cases} -1 & \text{if } x = 0 \\ x & \text{else} \end{cases}$$

and let the classification prediction value $h_i(a)$ of a w.r.t. h, T and $\mathcal{K}$ be defined as (again for ease, we omit $\mathcal{K}$ and T)

$$h_i(a) = t(bed(\mathcal{K} \cup \{h_i\}, a{:}T)) \in \{-1\} \cup (0, 1] \ .$$

We also define the examples labelling l over I in the following way: for $a \in \mathsf{I}$

$$l(a) = \begin{cases} 1 & \text{if } (a, 1) \in \mathcal{E}^+ \\ -1 & \text{else} \ . \end{cases}$$

Then, the FUZZY OWL-BOOST algorithm calling iteratively a weak learner is shown in 1, which we comment briefly next. The algorithm is essentially the same as $\mathbb{R}$eal AdaBoost, except for few context dependent parts. In Step 2 we initialise the set of individuals I to be considered as $\mathsf{I}_{\mathcal{K}}$. Essentially, all individuals will be weighted. The main loop (Steps 5 - 11) is the same as for $\mathbb{R}$eal AdaBoost with the particularity that Step 6 we invoke a fuzzy GCI (weak) learner that is assumed to return a GCI of the form $\langle C_i \sqsubseteq T, d_i \rangle$. Note that, for ease of presentation, we didn't include an additional condition that causes a

[14] The weight of $a \in \mathsf{I}$ w.r.t. $\mathbf{w}$ is denoted w_a.

Algorithm 1 FUZZY OWL-BOOST

Input: KB $\mathcal{K}$, training set $\mathcal{E}$, target concept name T, number of iterations n

Output: Hypothesis $\mathcal{H}$ (Fuzzy OWL EL TBox) of the form of Eq. 6

1: $\mathcal{H} \leftarrow \emptyset$;
2: $\mathsf{I} \leftarrow \mathsf{I}_{\mathcal{K}}$;
3: $\mathbf{w}_1 \leftarrow \mathbf{u}$; ▷ Initialise the weight distribution over I
4: // Main boosting loop
5: **for** $i = 1$ to n **do**
6: $\quad h_i \leftarrow$ FUZZYWEAKLEARNER($\mathcal{K}$, T, $\mathcal{E}$, $\mathbf{w}_i$); ▷ h_i is of the form $\langle C_i \sqsubseteq T, d_i \rangle$
7: $\quad h^\star_i \leftarrow \max_{a \in \mathsf{I}} |h_i(a)|$; ▷ $h^\star_i$ is the maximal value of h_i over I
8: $\quad \mu_i \leftarrow \frac{1}{h^\star_i} \sum_{a \in \mathsf{I}} w_{i,a} \cdot l(a) \cdot h_i(a)$; ▷ μ_i is the *normalised* margin of h_i w.r.t. I
9: $\quad \alpha_i \leftarrow \frac{1}{2h^\star_i} \cdot \ln \frac{1+\mu_i}{1-\mu_i}$; ▷ α_i is the weight of classifier $h_i$in the ensemble
10: $\quad$ **for all** $a \in \mathsf{I}$ **do** ▷ Update the weight distribution
11: $\quad\quad w_{i+1,a} \leftarrow w_{i+1,a} \cdot (\frac{1-(\mu_i \cdot l(a) \cdot h_i(a))/h^\star_i}{1-\mu_i^2})$;
12: $\quad \mathcal{H} \leftarrow \mathcal{H} \cup \{\langle C_i \sqsubseteq T_i, d_i \rangle \mid T_i \text{ new}\}$
13: // Build now the final classifier ensemble
14: **for** $i = 1$ to n **do** ▷ Normalise the α_i via the softmax function
15: $\quad \beta_i \leftarrow \frac{e^{\alpha_i}}{\sum_{j=1}^{n} e^{\alpha_i}}$;
16: $D \leftarrow \beta_1 \cdot C_1 + \ldots \beta_n \cdot C_n$
17: $d \leftarrow cf(D \sqsubseteq T, \mathsf{I})$; ▷ d is the classifier ensemble confidence degree
18: $h \leftarrow \langle \beta_1 \cdot T_1 + \ldots \beta_n \cdot d_n \cdot T_n \sqsubseteq T, d \rangle$; ▷ The final classifier ensemble
19: $\mathcal{H} \leftarrow \mathcal{H} \cup \{h\}$;
20: **return** $\mathcal{H}$;

break of the loop. In fact, an implicit condition of boosting is that the error of a weak learner is below 0.5. This may implemented in our case by adding another step before Step 12 that computes the error

$$\epsilon = \sum_{a \in \mathsf{I}} \delta(h_i(a), l(a)) \cdot h_i(a) \ ,$$

where $\delta(x, y) \in \{0, 1\}$ is defined as $(x \in \{-1\} \cup (0, 1], y \in \{-1, 1\})$

$$\delta(x, y) = \begin{cases} 1 & \text{if } x \cdot y < 0 \\ 0 & \text{else} \end{cases}$$

and determines whether there is a *disagreement* among the sign of $h_i(a)$ and $l(a)$. Then, if $\epsilon \geq 0.5$ we break the loop. In Step 12 we add the (weak) learned fuzzy GCI $\langle C_i \sqsubseteq T, d_i \rangle$ to the hypothesis set $\mathcal{H}$. In Steps 14 - 18 we prepare the final classifier ensemble. To do so, we have to perform a normalisation step. In fact, since in Real AdaBoost generally $\alpha_i \in \mathbb{R}$, we have to normalise the set of values α_i $(1 \leq i \leq n)$ before building the weighted sum in Step 16. To do so, we rely on the well-known softmax function. Eventually, in Step 17, we determine the degree to be attached to the ensemble classifier computed as the confidence value, which resembles the well-known *precision* measure used in macchine learning. [15]

We next describe the weak learner we employ here. As anticipated, will use a FOIL-$\mathcal{DL}$ [67, 68, 70] like weak learner, which however needs to be adapted to our specific setting. In general terms the weak learning algorithm, called wFOIL-$\mathcal{DL}$, proceeds as follows:

1. start from concept $\top$;
2. apply a refinement operator to find more specific concept description candidates;
3. exploit a scoring function to choose the best candidate;
4. re-apply the refinement operator until a good candidate is found;

[15] Precision is also called *positive predictive value* and roughly is the percentage of positive instances among all retrieved instances.

Table 2: Downward Refinement Operator.

$$\rho(C) = \begin{cases} \mathbf{A}_{\mathcal{K}} \cup \{\exists R.\top \mid R \in \mathbf{R}_{\mathcal{K}}\} \cup \{\exists S.d \mid S \in \mathbf{S}_{\mathcal{K}}, d \in \mathcal{D}\} \cup & \\ \qquad \{\exists S.\mathbf{true}, \exists S.\mathbf{false}, \mid S \in \mathbf{B}_{\mathcal{K}}\} & \text{if } C = \top \\ \{A' \mid A' \in \mathbf{A}_{\mathcal{K}}, \mathcal{K} \models A' \sqsubseteq A\} \cup \{A \sqcap A'' \mid A'' \in \rho(\top)\} & \text{if } C = A \\ \{\exists R.D' \mid D' \in \rho(D)\} \cup \{(\exists R.D) \sqcap D'' \mid D'' \in \rho(\top)\} & \text{if } C = \exists R.D, R \in \mathbf{R}_{\mathcal{K}} \\ \{(\exists S.d) \sqcap D \mid D \in \rho(\top)\} & \text{if } C = \exists S.d, S \in \mathbf{S}_{\mathcal{K}}, d \in \mathcal{D} \\ \{(\exists S.d) \sqcap D \mid D \in \rho(\top)\} & \text{if } C = \exists S.d, S \in \mathbf{B}_{\mathcal{K}}, d \in \{\mathbf{true}, \mathbf{false}\} \\ \{C_1 \sqcap ... \sqcap C_i' \sqcap ... \sqcap C_n \mid i = 1, ..., n, C_i' \in \rho(C_i)\} & \text{if } C = C_1 \sqcap ... \sqcap C_n \end{cases}$$

5. iterate the whole procedure until a satisfactory coverage of the positive examples is achieved.

We briefly detail these steps.

Computing fuzzy datatypes. For a numerical datatype S, we allow *equal width triangular/trapezoidal partition* of values $V_S = \{v \mid \mathcal{K} \models a{:}\exists S. =_v\}$ into a finite number of fuzzy sets (typically, 3 or 5 sets), which is identical to [67, 70, 114] (see, *e.g.* Figure 2). However, we additionally, allow also the use of the C-means fuzzy clustering algorithm over V_S with 3 or 5 clusters, where the fuzzy membership function is a triangular function build around the centroid of a cluster. Note that C-means has not been considered in [67, 70, 114]. [16]

The refinement operator. The refinement operator we employ is the same as in [67, 68, 74, 114] except that now we add the management of boolean values as well. Essentially, the refinement operator takes as input a concept C and generates new, more specific concept description candidates D (*i.e.*, $\mathcal{K} \models D \sqsubseteq C$). For the sake of completeness, we recap the refinement operator here. Let $\mathcal{K}$ be an ontology, $\mathbf{A}_{\mathcal{K}}$ be the set of all atomic concepts in $\mathcal{K}$, $\mathbf{R}_{\mathcal{K}}$ the set of all object properties in $\mathcal{K}$, $\mathbf{S}_{\mathcal{K}}$ the set of all numeric datatype properties in $\mathcal{K}$, $\mathbf{B}_{\mathcal{K}}$ the set of all boolean datatype properties in $\mathcal{K}$ and $\mathcal{D}$ a set of (fuzzy) datatypes. The refinement operator ρ is shown in Table 2.

The scoring function. The scoring function we use to assign a score to each candidate hypothesis is essentially a weighted *gain* function, similar to the one employed in [67, 68, 74, 114] and implements an information-theoretic criterion for selecting the best candidate at each refinement step. Specifically, given a GCI ϕ of the form $C \sqsubseteq T$ chosen at the previous step, a KB $\mathcal{K}$, a set of individuals I, a weight distribution $\mathbf{w}$ over I, a set of examples $\mathcal{E}$ and a candidate GCI ϕ' of the form $C' \sqsubseteq T$, then

$$gain(\phi', \phi, \mathbf{w}, \mathsf{I}) = p * (log_2(cf(\phi', \mathbf{w}, \mathsf{I})) - log_2(cf(\phi, \mathbf{w}, \mathsf{I}))) \ , \tag{8}$$

where $p = |C' \sqcap C|_{\mathcal{K}}^{\mathbf{w}, \mathsf{I}_{\mathcal{E}^+}}$ is the weighted cardinality of positive examples covered by ϕ that are still covered by ϕ'. Note that the gain is positive if the confidence degree increases.

Stop Criterion. wFoil-$\mathcal{DL}$ stops when the confidence degree is above a given threshold $\theta \in [0, 1]$, or no better weak learner can be found that does not cover any negative example (in $\mathcal{E}^-$) above a given percentage. Note that in Foil-$\mathcal{DL}$ instead, non-positive examples are not allowed to be covered.

The wFoil-$\mathcal{DL}$ Algorithm. The wFoil-$\mathcal{DL}$ algorithm is defined in Algorithm 2, which we comment briefly as next. Steps 1 - 3 are simple initialisation steps. Steps 5 - 21. are the main loop from which we may exit in case there is no improvement (Step. 16), and the confidence degree of the so far determined weak learner is above a given threshold or it does not cover any negative example above a given percentage (Step. 18). Note that the latter case guarantees soundness of the weak learner if the percentage is set to 0. In Step 8 we determine all new refinements, which then are scored in Steps 10 -15 in order to determine the one with the best gain. Eventually, once we exit from the main loop, the best found weak learner is returned (Step 22 and 23).

Remark 6 *As for* Foil-$\mathcal{DL}$ *(and* pFoil-$\mathcal{DL}$*), the weak learner* wFoil-$\mathcal{DL}$ *also allows to use a backtracking mechanism (Step 20.), which, for ease of presentation, we omit to include. The mechanism is exactly the same as for the* pFoil-$\mathcal{DL}$*-learnOneAxiom described in [114, Algorithm 3]. Essentially, a stack of top-k refinements is maintained, ranked in decreasing order of the confidence degree from which we pop the next best refinement (if the stack is not empty) in case no improvement has occurred in Step 19.* C_{best} *becomes the popped-up refinement.*

[16] Specifically, C-means has not been considered so far in fuzzy GCI learning.

Algorithm 2 wFOIL-$\mathcal{DL}$

Input: KB $\mathcal{K}$, target concept name T, training set $\mathcal{E}$, weight distribution $\mathbf{w}$, confidence threshold $\theta \in [0, 1]$, non-positive coverage percentage $\eta \in [0, 100]$

Output: Weak hypothesis of the form $\langle C \sqsubseteq T, d \rangle$

$\mathsf{I} \leftarrow \mathsf{I}_{\mathcal{K}}$;
$C \leftarrow \top$; ▷ Start from $\top$
$\phi \leftarrow C \sqsubseteq T$;
//Loop until no improvement
while $C \neq$ **null do**
    $C_{best} \leftarrow C$;
    $maxgain \leftarrow 0$;
    $\mathcal{C} \leftarrow \rho(C)$; ▷ Compute all refinements of C
    // Compute the score of the refinements and select the best one
    for all $C' \in \mathcal{C}$ **do**
        $\phi' \leftarrow C' \sqsubseteq T$;
        $gain \leftarrow gain(\phi', \phi, \mathbf{w}, \mathsf{I})$;
        if $(gain > maxgain)$ **and** $(cf(\phi', \mathbf{w}, \mathsf{I}) > cf(\phi, \mathbf{w}, \mathsf{I}))$ **then**
            $maxgain \leftarrow gain$;
            $C_{best} \leftarrow C'$;
    if $C_{best} = C$ **then** ▷ No improvement
        //Stop if confidence degree above threshold or no negative coverage below threshold
        if $(cf(C_{best} \sqsubseteq T, \mathsf{I}) \geq \theta)$ **and** $\frac{\lceil C_{best} \rceil_{\mathcal{K}}^{\mathsf{I}_{\mathcal{E}^-}}}{|\mathsf{I}_{\mathcal{E}^-}|} \leq \eta$ **then break**;
        // Manage backtrack here, if foreseen
    $C \leftarrow C_{best}$;
    $\phi \leftarrow C \sqsubseteq T$;
$d \leftarrow cf(\phi, \mathsf{I})$; ▷ Compute the weak classifier confidence degree
return $\langle \phi, d \rangle$;

Table 3: Facts about the ontologies of the experiment.

ontology	**DL**	**class.**	**obj. prop.**	**data. prop.**	**ind.**	**target** T	pos	neg	max d./c./fp
FamilyTree	$\mathcal{SROIF}(\mathcal{D})$	22	52	6	368	Uncle	46	156	1/5/0
Hotel	$\mathcal{ALCOF}(\mathcal{D})$	89	3	1	88	Good_Hotel	12	11	1/5/0
Moral	$\mathcal{ALC}$	46	0	0	202	ToLearn_Guilty	102	100	1/5/0
SemanticBible (NTN)	$\mathcal{SHOIN}(\mathcal{D})$	51	29	9	723	ToLearn_Woman	46	3	1/5/0
UBA	$\mathcal{SHI}(\mathcal{D})$	44	26	8	1268	Good_Researcher	22	113	1/5/0
WineOnto	$\mathcal{SHI}(\mathcal{D})$	178	15	7	138	ToLearn_DryWine	15	-	1/5/0
Pair50	$\mathcal{ALC}$	3	6	0	311	ToLearn	20	29	2/5/0
Straight	$\mathcal{ALC}$	3	6	0	347	ToLearn	4	50	3/5/100
Lymphography	$\mathcal{ALC}$	50	0	0	148	ToLearn	81	67	1/5/100
Mammographic	$\mathcal{ALC}(\mathcal{D})$	20	3	2	975	ToLearn	445	516	3/5/100
Pyrimidine	$\mathcal{ALC}(\mathcal{D})$	2	0	27	74	ToLearn	20	20	1/5/100
Suramin	$\mathcal{ALC}(\mathcal{D})$	47	3	1	2979	ToLearn	7	10	3/5/100

Table 4: Datasets considered from the UCI ML Repository.

dataset	instances	attributes	target T	pos
Iris	151	4	Iris-setosa Iris-versicolor Iris-virginica	51 50 50
Wine	178	13	$1, 2, 3$	$59, 71, 48$
Wine Quality	4898	12	GoodRedWine	18
Yeast	1484	8	CYT, ERL, EXC, ME1, ME2 ME3, MIT, NUC, POX, VAC	444, 5, 35, 44, 51 163, 244, 426, 20, 30

5 Evaluation

We have implemented the algorithm within the *FuzzyDL-Learner* [17] system and evaluated it. All the data and implementation can be downloaded from the FuzzyDL-Learner home page. If not specified differently, in all runs we used the `HermiT` [18] OWL reasoner. Sometimes we also used the `jFact` [19] reasoner, indicated with (`jF`) in the result tables.

5.1 Setup

A number of OWL ontologies from different domains have been selected as illustrated in Tables 3 and 4. Note that the ontologies in Table 4 are not available as OWL 2 ontologies and, thus, we have translated them from a a csv format according to the procedure shown in Section 5.2.

For each ontology $\mathcal{K}$ a meaning full target concept has been selected such that the conditions of the learning problem are satisfied. We report also the DL the ontology refers to, the number of concept names, object properties, datatype properties and individuals in the ontology. We also report the maximal nesting depth (max d.), maximal number of conjuncts (max c.) and maximal percentage of false positives (max fp) during the learning phase. The number n of iterations of Fuzzy OWL-Boost is set to 10. We didn't consider backtracking. Nevertheless, all configuration parameters for each run are available from the downloadable data.

We will consider the following performance indices (see also [114] for similar measures), which we report here for clarity to avoid ambiguity. So, consider the learned fuzzy GCI of the form $\langle D \sqsubseteq T, d\rangle$, where D and d are determined by Fuzzy OWL-Boost in Steps 16 and 17, respectively, and the classifier ensemble $\mathcal{H}$ returned by Fuzzy OWL-Boost. [20] Then

Fuzzy True Positives: denoted TP_f, is defined as

$$TP_f = |D|^{\mathcal{E}^+}_{\mathcal{K}} , \tag{9}$$

Fuzzy False Positives: denoted FP_f, is defined as

$$FP_f = |D|^{\mathcal{E}^-}_{\mathcal{K}} , \tag{10}$$

[17] http://www.umbertostraccia.it/cs/software/FuzzyDL-Learner/.

[18] http://www.hermit-reasoner.com

[19] http://jfact.sourceforge.net

[20] Note that the two are the same from a classification point of view.

Fuzzy True Non-Positive: denoted TNP_f, is defined as

$$TNP_f = |\mathsf{I}_{\mathcal{E}^-}| - FP_f \ , \tag{11}$$

Fuzzy False Non-Positive: denoted FNP_f, is defined as

$$FNP_f = |\mathsf{I}_{\mathcal{E}^+}| - TP_f \ , \tag{12}$$

Fuzzy Precision: denoted P_f, is defined as

$$P_f = \frac{TP_f}{|D|_{\mathcal{K}}^{\mathsf{I}_{\mathcal{E}}}} = cf(D \sqsubseteq T, \mathsf{I}_{\mathcal{E}}) = d \ , \tag{13}$$

Fuzzy Recall: denoted R_f, is defined as

$$R_f = \frac{TP_f}{|\mathsf{I}_{\mathcal{E}^+}|} \ , \tag{14}$$

Fuzzy $F1$-score: denoted $F1_f$, is defined as

$$F1_f = 2 \cdot \frac{P_f \cdot R_f}{P_f + R_f} \ ,$$

Mean Squared Error: denoted MSE, is defined as

$$MSE = \frac{1}{|\mathsf{I}_{\mathcal{E}}|} \cdot \sum_{a \in \mathsf{I}_{\mathcal{E}}} (\mathcal{H}(a) - \mathcal{E}(a))^2 \ ,$$

where $\mathcal{H}(a) \in [0,1]$ is the *classification prediction value* of a w.r.t. $\mathcal{H}$, T, which is defined as

$$\mathcal{H}(a) = bed(\mathcal{K} \cup \mathcal{H}, a{:}T) \ .$$

Standard fuzzy logic has been chosen to compute the indices. Concerning the fuzzy measures above, we consider also their well-known crips variants [6] (in the denotation we omit the f subscript), obtained by replacing in the equations above the cardinality function $|\cdot|_{\mathcal{K}}^{\mathsf{I}_{\mathcal{K}}}$ (see Eq. 1) with the crisp cardinality function $\lceil\cdot\rceil_{\mathcal{K}}^{\mathsf{I}_{\mathcal{K}}}$.

A k-fold cross validation design was adopted (specifically, $k = 5$) to determine the average of the above described performance indices. For each measure, the (macro) average value over the various folds is reported in the tables. In all tests, we have that $\mathsf{I}_{\mathcal{E}} = \mathsf{I}_{\mathcal{K}}$ and that there is at least one positive example in each fold, while the other examples of a fold have been randomly been selected. Eventually, we considered also the extreme case in which the whole set $\mathcal{E}$ is used for both learning and testing. This case has been considered for those ontologies with few positive examples for which k-fold cross validation is not meaningful and also for the task aiming at "explaining" the target w.r.t. the given data set. This case is indicated with $\star$ in the results tables. As baseline, we considered FOIL-$\mathcal{DL}$, which learns, conceptually, rules sets of the form (compare with Eq. 6)

$$\begin{array}{l} \langle C_1 \sqsubseteq T_1, d_1 \rangle \\ \quad\vdots \\ \langle C_n \sqsubseteq T_n, d_n \rangle \\ \langle T_1 \sqcup \ldots \sqcup T_n \sqsubseteq T, d \rangle \end{array} \tag{15}$$

In the result Tables 5-7, for a given ontology, algorithm and clustering method (uniform or C-means [21]), we report only the run with the best effectiveness measure determined by

$$best = (1 - MSE) \cdot F1 \ , \tag{16}$$

[21]For C-means, we fixed the hyper-parameter to $m = 0.5$, the threshold to $\epsilon = 0.05$ and number of maximum iterations to 100.

which minimises the MSE and maximises $F1$ with the threshold $\theta \in \{0.04, 0.34, 0.64, 0.94, 1.0\}$ and number of fuzzy sets $c \in \{3, 5, 7\}$. For each table entry, we report also the average number ($\#r$) of rules learned and the average rule body length (l) of a learned rule computed as

$$|C| = \begin{cases} 1 & \text{if } C = A \\ 1 + 2 \cdot |D| & \text{if } C = \exists R.D \\ |C_1| + |C_2| + \ldots + |C_n| & \text{if } C = C_1 \sqcap C_2 \ldots \sqcap C_n \\ n & \text{if } C = \alpha_1 \cdot T_1 + \ldots \alpha_n \cdot T_n \\ n & \text{if } C = \max(T_1, \ldots, T_n)\ . \end{cases}$$

The averages $\#r$ and l are computed as macro averages among the folds. The intuition of $|\cdot|$ is that the less rules are leaned, the shorter and less nested they are the easier a rule set is to be interpreted by a human being. Note that for FOIL-$\mathcal{DL}$ (resp. for FUZZY OWL-BOOST) we do count the final aggregation rule as well, except for the case in which the set of of learned rules is 2, *i.e.* the aggregation rule body contains one concept name only ($n = 1$ in Eqs. 6 and 15). For both FOIL-$\mathcal{DL}$ and FUZZY OWL-BOOST we report in red the run with the maximal value of *best* and then compare these two runs w.r.t. $\#r$ and l and report in red the one with the better $\#r \cdot l$ score.

5.2 UCI ML conversion algorithm

To start with, we conducted a preliminary experiment that focuses on fuzzy set construction involved in numerical fuzzy datatypes, *i.e.* about partitioning the data into 3, 5 or 7 fuzzy sets, considering uniform partitioning (see Fig. 2) or partitioning via C-means. [22] To this purpose, we considered the well-known *UC Irvine Machine Learning Repository* [30] from which selected some popular datasets with numerical attributes as shown in Table 4. While evaluating ontology-based learning algorithms is untypical on those data sets [23], we believe it is still interesting to to do so as a main ingredient of our algorithm is the use of fuzzy concrete datatype properties.

As anticipated, as the the data sets in Table 4 are not available as OWL 2 ontologies, we have translated them from a csv format a into an OWL 2 ontology in a simple way, which we describe next. The method is quite general and can be applied to any other dataset with similar specification and a dedicated procedure is available within our implemented learner for future evaluations.

Consider a dataset D with (functional) attributes $S_1, \ldots, S_n$ of type $t_1, \ldots, t_n$. Each data record r is of then of the form $\langle v_1, \ldots, v_n, T \rangle$, where v_i is the value of attribute S_i of type t_i, while T is the target class name for record r. For instance, for the iris dataset we have attributes

$$\texttt{sepal\_length}, \texttt{sepal\_width}, \texttt{petal\_length}, \texttt{petal\_width}$$

of type

$$\texttt{double}, \texttt{double}, \texttt{double}, \texttt{double}$$

and the first record r is

$$\langle \texttt{5.1}, \texttt{3.5}, \texttt{1.4}, \texttt{0.2}, \texttt{Iris} - \texttt{setosa} \rangle\ .$$

The knowledge base $\mathcal{K}_D = \langle \mathcal{T}_D, \mathcal{A}_D \rangle$ built to describe the data is as follows. Let T_D be the set of all target class names T occurring in D. The TBox $\mathcal{T}_D$ is

$$\begin{array}{lll} T & \sqsubseteq & \texttt{class} \quad (T \in T_D) \\ \texttt{class} & \sqsubseteq & \exists S_i.t_i \quad (i = 1 \ldots n)\ . \end{array} \tag{17}$$

Additionally, each data property S_i has been declared as functional.

The ABox $\mathcal{A}_D$ is built in the following way. For each record r of the form $\langle v_1, \ldots, v_n, T \rangle$, we create a new individual a_r and add the axioms

$$\begin{array}{l} a_r{:}T \\ a_r{:}\exists S_i. =_{v_i} \quad (i = 1 \ldots n) \end{array} \tag{18}$$

[22] Let us also recall that in ontology-based learning rarely datatype properties have been have considered.

[23] To the best of our knowledge, we are unaware of any evaluation of ontology-based methods on those data sets.

to $\mathcal{A}_D$. For instance, for the iris dataset described above, which has three target classes $\texttt{Iris}-\texttt{setosa}$, $\texttt{Iris}-\texttt{versicolor}$ and $\texttt{Iris}-\texttt{virginica}$, the KB contains the axioms

$$
\begin{array}{l}
\texttt{Iris}-\texttt{setosa} \sqsubseteq \texttt{class} \\
\texttt{Iris}-\texttt{versicolor} \sqsubseteq \texttt{class} \\
\texttt{Iris}-\texttt{virginica} \sqsubseteq \texttt{class} \\
\texttt{class} \sqsubseteq \exists \texttt{sepal\_length}.\mathbf{double} \\
\texttt{class} \sqsubseteq \exists \texttt{sepal\_width}.\mathbf{double} \\
\texttt{class} \sqsubseteq \exists \texttt{sepal\_length}.\mathbf{double} \\
\texttt{class} \sqsubseteq \exists \texttt{sepal\_width}.\mathbf{double} \\
a_1{:}\texttt{Iris}-\texttt{setosa} \\
a_1{:}\exists \texttt{sepal\_length}.=_{5.1} \\
a_1{:}\exists \texttt{sepal\_width}.=_{3.5} \\
a_1{:}\exists \texttt{petal\_length}.=_{1.4} \\
a_1{:}\exists \texttt{sepal\_width}.=_{0.2}\ .
\end{array}
$$

It is easily verified that the KB $\mathcal{K}_D$ constructed for each dataset D (i) belongs to the DL $\mathcal{ALEF}(\mathbf{D})$; (ii) the number of classes is $|T_D| + 1$; and there are n functional datatype properties.

For each dataset D, we run a k-fold cross validation on ontologies $\mathcal{K}_D$ for each target class $T \in T_D$, by varying the number of fuzzy sets $(3, 5$ or $7)$, selecting uniform partition or C-means, and varying the confidence threshold $\theta \in \{0.04, 0.34, 0.64, 0.94, 1.0\}$. Non-positive coverage percentage is set to $\eta = 0.0$ and the numbers of boosting iterations is $n = 10$. Maximal number of conjuncts is set to 5, while maximal nesting depth is set, of course, to 1. We report the parameters used obtained through our grid search through varying parameters, maximising *best* (see Eq. 16) (computed as macro average among the folds).

As an analytic per problem would require a lot of space, we report the results succinctly in Tables 5 and 6. Nevertheless, all parameters and results are reported in the downloadable package.

5.3 Succinct discussion

Consider the values in red in Tables 5 and 6 (the UCI ML data set). As we can see, concerning $F1$, there seem to be no substantial difference among the best runs for Foil-$\mathcal{DL}$ and Fuzzy OWL-Boost. Surprisingly, there is also no clear winner among uniform clustering (case u) and C-means (case c). However, interestingly in most cases Fuzzy OWL-Boost learns a smaller ensemble (in terms of $\#r \cdot l$) w.r.t. Foil-$\mathcal{DL}$ (12 wins vs. 4 wins), which roughly means that Fuzzy OWL-Boost learns in general easier human interpretable rule sets than Foil-$\mathcal{DL}$. Also note that in one case, `Yeast - ERL`, Foil-$\mathcal{DL}$ was unable to learn a rule set while Fuzzy OWL-Boost was instead.

Interestingly, a similar analysis applies also to the red results in Table 7 with a notable exception concerning the last four cases `Lymphography - Suramin` in which Fuzzy OWL-Boost learns striking smaller ensembles than Foil-$\mathcal{DL}$. We suspect that Foil-$\mathcal{DL}$ produces here larger sets as it removes covered positives and then focuses on the remaining ones producing in this way specialised rules that cover few positive examples only.

In summary, while there seems no clear winner among uniform clustering vs. C-means, and Fuzzy OWL-Boost vs. Foil-$\mathcal{DL}$ in therms of $F1$, Fuzzy OWL-Boost seems to learn smaller and, thus, easier human interpretable ensembles than Foil-$\mathcal{DL}$, which is of utmost importance if the rules sets need to be analysed by an expert of the field.

6 Conclusions & Future Work

In this work, we addressed the problem of automatically learning fuzzy concept inclusion axioms from OWL 2 ontologies. That is, given a target class T of an OWL ontology, we address the problem of inducing fuzzy $\mathcal{EL}(\mathbf{D})$ concept inclusion axioms that describe sufficient conditions for being an individual instance of T. In particular, we have adapted the $\mathbb{R}$eal AdaBoost [86] boosting algorithm to the fuzzy OWL case, by presenting the Fuzzy OWL-Boost algorithm. The main features of our algorithm are essentially the fact that (i) it generates a set of fuzzy fuzzy $\mathcal{EL}(\mathbf{D})$ inclusion axioms, which are the weak hypothesis, possibly including fuzzy concepts and fuzzy concrete domains; and (ii) all generated fuzzy concept inclusion axioms can be directly encoded as *Fuzzy OWL 2* axioms.

Table 5: UCI ML Repository results.

dataset & target		param.				Foil							param.				Fuzzy OWL-Boost						
	part.	θ	f. s.	#r	l	MSE	P_f	R_f	$F1_f$	P	R	$F1$	θ	f. s.	#r	l	MSE	P_f	R_f	$F1_f$	P	R	$F1$
Iris (setosa)	u	0.94	3	3.8	3.08	0.0142	1.00	0.8082	0.8934	1.0	1.0	**1.0**	1.0	3	**1**	**1.24**	0.019	1	0.8026	0.8898	1.0	1.0	**1.0**
	c	0.64	7	1	2	0.0028	0.9898	0.9516	0.9702	0.9136	1.0	0.954	0.64	7	1.8	3.73	0.0555	1.0	0.658	0.79175	1.0	1.0	1.0
Iris (versicolor)	u	0.94	7	3.8	3.08	0.0982	0.9658	0.5054	0.6568	0.833	0.96	0.8892	0.94	3	1	4	0.128	0.916	0.392	0.549	0.833	1.0	0.909
	c	0.94	7	4.4	3.84	0.0902	0.9714	0.5768	0.7096	0.9134	0.88	**0.8884**	0.64	3	**1**	**4**	0.0464	0.9106	0.6804	0.777	0.7606	1.0	**0.86**
Iris (virginica)	u	0.94	5	**1.4**	**2**	0.0482	0.955	0.7694	0.8508	0.8588	0.94	**0.8912**	1.0	7	3	2.66	0.1404	1	0.4086	0.58	1.0	0.82	0.8992
	c	0.64	7	4.2	2.76	0.0774	0.8932	0.6722	0.7628	0.7966	1	0.8824	0.64	3	1.4	3.87	0.062	0.926	0.6784	0.7798	0.748	0.98	**0.8428**
Wine (1)	u	0.94	5	4.6	3.86	0.0764	0.9392	0.615	0.7416	0.8764	0.9668	**0.9154**	0.94	3	**2**	**2.5**	0.101	0.9553	0.54	0.6847	0.9047	0.9697	**0.9327**
	c	0.94	7	7.6	4.2	0.1076	0.9266	0.5226	0.66	0.9288	0.892	0.9162	0.64	3	4.2	4.92	0.1216	0.969	0.4492	0.6082	0.8468	0.9834	0.9082
Wine (2)	u	0.94	3	4	4.1	0.1514	0.9612	0.4412	0.5886	0.8646	0.9306	0.8876	0.64	7	7.8	4.82	0.2548	0.9132	0.2134	0.3444	0.8226	0.9582	0.876
	c	0.94	5	6.4	3.85	0.0634	0.9522	0.7644	0.8418	0.876	0.9438	**0.902**	0.64	3	**5**	**4.88**	0.1278	0.9436	0.4874	0.641	0.8128	0.9716	**0.8772**
Wine (3)	u	0.94	7	**4.2**	**2.94**	0.0562	0.9734	0.607	0.7468	0.9418	1.0	**0.9692**	0.94	7	4	3.82	0.1472	0.952	0.28	0.4316	0.9396	0.9578	0.9482
	c	0.94	5	4.4	3.93	0.0572	0.9448	0.6982	0.7992	0.9132	0.9556	0.928	1.0	5	3	5.93	0.1222	0.9968	0.4104	0.5738	0.9578	0.8978	**0.9214**
Wine Quality	u	0.04	7	3.5	3.75	0.003	0.058	0.031	0.039	0.0345	0.6115	0.0655	0.04	5	1	8	0.003	0.0495	0.0045	0.0085	0.0315	0.111	0.049
	c	0.04	7	4.5	4.4	0.003	0.0515	0.0395	0.0435	0.035	0.778	**0.067**	0.04	5	**1**	**5**	0.003	0.055	0.034	0.042	0.079	0.667	**0.141**

Table 6: UCI ML Repository results: **Yeast**

target	part.	param. θ	param. f. s.	#r	l	Foil MSE	P_f	R_f	$F1_f$	P	R	$F1$
CYT	u	0.34	7	**3.80**	**2.52**	0.2112	0.392	0.2368	0.295	0.3696	0.9706	**0.5354**
	c	0.34	7	7.80	4.47	0.2316	0.3616	0.2026	0.2588	0.358	0.8828	0.5092
ERL	u	–	–	–	–	–	–	–	–	–	–	–
	c	–	–	–	–	–	–	–	–	–	–	–
EXC	u	0.34	5	**2.20**	**4.40**	0.0192	0.3672	0.1368	0.1988	0.2806	0.7142	**0.4008**
	c	0.34	5	1.67	6.89	0.0244	0.0138	0.0048	0.0072	0.0572	0.1142	0.0762
ME1	u	0.94	7	4.40	5.04	0.023	0.7126	0.1554	0.2406	0.6148	0.5972	**0.5458**
	c	0.34	7	2.80	4.17	0.0212	0.495	0.256	0.336	0.36	0.7056	0.4736
ME2	u	0.34	7	4.20	4.38	0.0292	0.375	0.1226	0.183	0.2842	0.589	**0.3804**
	c	0.34	5	1.40	5.73	0.298	0.4574	0.107	0.1696	0.3224	0.4146	0.3694
ME3	u	0.64	7	4.80	4.19	0.0704	0.8186	0.255	0.385	0.7502	0.7414	0.7438
	c	0.64	7	6	4.85	0.0604	0.8444	0.3484	0.4916	0.7532	0.7784	**0.7614**
MIT	u	0.64	5	4	3.85	0.124	0.6842	0.1846	0.2904	0.5952	0.5448	0.5686
	c	0.34	5	3.60	3.62	0.1168	0.5964	0.2574	0.3578	0.5122	0.652	**0.5708**
NUC	u	0.34	7	6.60	3.74	0.21	0.3816	0.229	0.282	0.353	0.9036	0.503
	c	0.34	7	11.20	4.21	0.2084	0.4278	0.2496	0.314	0.363	0.9082	**0.5182**
POX	u	–	–	–	–	–	–	–	–	–	–	–
	c	–	–	–	–	–	–	–	–	–	–	–
VAC	u	–	–	–	–	–	–	–	–	–	–	–
	c	0.34	7	2.25	5.73	0.021	0.001	0.0002	0.0004	0.0126	0.0334	**0.0182**

target	part.	param. θ	param. f. s.	#r	l	Fuzzy OWL-Boost MSE	P_f	R_f	$F1_f$	P	R	$F1$
CYT	u	0.34	7	6.40	5.55	0.2614	0.4742	0.079	0.1346	0.4034	0.901	**0.5554**
	c	0.34	3	6.40	6.82	0.2642	0.4556	0.0728	0.1252	0.394	0.867	0.5408
ERL	u	–	–	–	–	–	–	–	–	–	–	–
	c	0.34	5	**1**	**10**	0.003	1.0	0.155	0.268	1.0	0.25	**0.4**
EXC	u	0.34	7	2.25	6	0.0222	0.3155	0.041	0.0725	0.34875	0.4287	**0.383**
	c	0.34	5	1	8	0.022	0.804	0.05	0.095	0.714	0.179	0.286
ME1	u	0.34	7	3.40	5.68	0.0272	0.758	0.054	0.0982	0.7004	0.5972	0.5128
	c	0.34	7	**3**	**4.40**	0.0232	0.4592	0.1364	0.2072	0.3804	1.0	**0.5484**
ME2	u	0.34	7	**3.25**	**5.02**	0.032	0.4527	0.0465	0.0827	0.431	0.475	**0.4418**
	c	0.34	7	3.50	6.25	0.0335	0.3765	0.016	0.03	0.3365	0.25	0.2765
ME3	u	0.64	5	**5**	**4.40**	0.083	0.8	0.136	0.233	0.641	0.838	**0.727**
	c	0.64	7	9	6.18	0.101	0.866	0.0516	0.0968	0.8124	0.582	0.6708
MIT	u	0.34	3	**3**	**4.53**	0.1306	0.6666	0.1338	0.2222	0.503	0.6388	**0.561**
	c	0.34	5	8.20	6.79	0.153	0.787	0.0454	0.0852	0.7188	0.4214	0.5218
NUC	u	0.34	5	**9**	**4.89**	0.256	0.457	0.065	0.114	0.37	0.938	**0.531**
	c	0.34	3	5.80	5.69	0.281	0.528	0.0602	0.1058	0.4876	0.5682	0.4702
POX	u	–	–	–	–	–	–	–	–	–	–	–
	c	–	–	–	–	–	–	–	–	–	–	–
VAC	u	–	–	–	–	–	–	–	–	–	–	–
	c	0.34	5	**1**	**10**	0.021	1.0	0.02	0.04	1.0	0.042	**0.08**

Table 7: Results on Ontologies

target	part.	param. θ	param. f. s.	Foil #r	l	MSE	P_f	R_f	$F1_f$	P	R	$F1$
FamilyTree (jF)	u	1.0	5	4.60	4.06	0.0192	0.89	0.9778	0.9292	0.89	0.9778	**0.9292**
	c	1.0	3	4.60	4.13	0.0192	0.89	0.9778	0.9292	0.89	0.9778	0.9292
Hotel (⋆)	u	0.64	7	7	4.43	0.011	0.922	0.875	0.898	0.857	1.0	0.923
	c	1.0	7	7	4.71	0.007	1.0	0.887	0.94	1.0	1.0	**1.0**
Moral (jF)	-	1.0	-	**3**	**1.33**	0.0	1.0	1.0	1.0	1.0	1.0	**1.0**
SemanticBible(NTN)	u	0.64	3	6.80	4.54	0.046	0.6774	0.469	0.5274	0.6608	0.513	**0.5478**
	c	0.64	3	6.80	4.54	0.046	0.6774	0.469	0.5274	0.6608	0.513	**0.5478**
UBA (⋆)	u	1.0	7	**3**	**2.67**	0.005	1.0	0.62	0.765	1.0	1.0	**1.0**
	c	1.0	7	**3**	**2.67**	0.006	1.0	0.483	0.652	1.0	1.0	**1.0**
WineOnto (jF,⋆)	u	1.0	3	**1**	**2**	0.0	1.0	1.0	1.0	1.0	1.0	**1.0**
	c	1.0	3	**1**	**2**	0.0	1.0	1.0	1.0	1.0	1.0	**1.0**
pair50 (jF, ⋆)	-	1.0	-	**1**	**3**	1.0	1.0	1.0	1.0	1.0	1.0	**1.0**
straight (jF, ⋆)	-	0.34	-	1	32	0.007	0.4	0.4	0.4	0.4	1	**0.571**
Lymphography (jF)	-	0.34	-	18.40	3.35	0.159	0.8594	0.8398	0.848	0.8608	0.8514	**0.8548**
Mammographic (jF)	u	0.34	3	19	8.05	0.2216	0.788	0.4792	0.5928	0.7366	0.692	**0.71**
	c	0.34	3	45	9.17	0.2706	0.866	0.3852	0.528	0.7888	0.7552	0.7568
Pyrimidine (jF, ⋆)	u	0.34	7	9	4.22	0.053	0.931	0.631	0.752	0.87	1	0.93
	c	1.0	7	13	5.84	0.05	1	0.801	0.889	1	0.9	**0.947**
Suramin (jF, ⋆)	u	0.34	3	3	14.66	0.002	0.457	0.206	0.284	0.412	1.0	0.583
	c	0.34	3	3	13.33	0.002	0.453	0.218	0.294	0.412	1.0	**0.583**

target	part.	param. θ	param. f. s.	Fuzzy OWL-Boost #r	l	MSE	P_f	R_f	$F1_f$	P	R	$F1$
FamilyTree (jF)	u	1.0	5	**3.40**	**3.89**	0.0118	0.982	0.8104	0.8768	0.9378	0.9778	**0.9566**
	c	0.94	3	**3.40**	**3.89**	0.0118	0.982	0.8104	0.8768	0.9378	0.9778	**0.9566**
Hotel (⋆)	u	1.0	5	5	6.20	0.092	1.0	0.192	0.322	1.0	0.833	0.909
	c	1.0	3	**5**	**6.20**	0.087	1.0	0.218	0.359	1.0	0.833	**0.909**
Moral (jF)	-	1.0	-	**3**	**1.33**	0.0436	1.0	0.7478	0.8556	1.0	1.0	**1.0**
SemanticBible(NTN)	u	0.34	3	**5.20**	**5**	0.0404	0.6198	0.2518	0.354	0.4568	0.7134	**0.5516**
	c	0.34	3	5.20	5.07	0.0404	0.6198	0.2518	0.354	0.4568	0.7134	0.5516
UBA (⋆)	u	1.0	7	4	3	0.001	1.0	0.825	0.904	1.0	1.0	1.0
	c	1.0	7	**3**	**2.67**	0.006	1.0	0.496	0.663	1.0	1.0	**1.0**
WineOnto (jF,⋆)	u	1.0	3	**1**	**2**	0.0	1.0	1.0	1.0	1.0	1.0	**1.0**
	c	1.0	3	**1**	**2**	0.0	1.0	1.0	1.0	1.0	1.0	**1.0**
pair50 (jF, ⋆)	-	1.0	-	1	5	1.0	1.0	1.0	1.0	1.0	1.0	**1.0**
straight (jF, ⋆)	-	0.34	-	**1**	**17**	0.007	0.4	0.4	0.4	0.4	1	**0.571**
Lymphography (jF)	-	0.64	-	**6**	**2.80**	0.2476	0.91	0.3976	0.5374	0.8442	0.8258	**0.8076**
Mammographic (jF)	u	0.64	3	4.6	4.83	0.2838	0.777	0.2646	0.3826	0.7166	0.6896	0.6698
	c	0.64	7	**6.2**	**5.98**	0.2846	0.9164	0.2848	0.4244	0.78	0.708	**0.727**
Pyrimidine (jF, ⋆)	u	0.64	5	8	6.88	0.193	0.836	0.164	0.275	0.69	1.0	0.816
	c	0.64	3	**4**	**6.25**	0.131	0.93	0.365	0.525	0.938	0.75	**0.833**
Suramin (jF, ⋆)	u	0.34	3	**3**	**1.66**	0.002	0.634	0.27	0.379	0.412	1.0	**0.583**
	c	0.34	3	**3**	**1.66**	0.002	0.634	0.27	0.379	0.412	1.0	**0.583**

We have also conducted an extensive evaluation, comparing it with FOIL-$\mathcal{DL}$. Our evaluation shows that, while essentially there seems no clear winner among FUZZY OWL-BOOST vs. FOIL-$\mathcal{DL}$ in therms of effectiveness, FUZZY OWL-BOOST seems to learn, however, smaller and, thus, easier human interpretable ensembles than FOIL-$\mathcal{DL}$, which is important if the rules sets need to be analysed by an expert of the field.

Concerning future work, besides investigating about other learning methods, we envisage various aspects worth to be investigated in more detail: (*i*) so far, we noticed nor relevant differences among uniform clustering and C-means clustering algorithms used in building fuzzy datatypes. This is somewhat surprising and we would like to investigated that in more detail by considering various alternatives as well, as proposed recently in a Fuzzy Sets and Systems special issue on fuzzy clustering [1]. Moreover, we would like to cover more OWL datatypes than those considered here so far (numerical and boolean) such as strings, dates, etc. possibly in combination with some sub-atomic classical machine learning methods (see, *e.g.* [110]); (*ii*) another aspect we would like to address in more detail is about the human interpretability of a rule set, *i.e.* to figure out learning algorithms that are more prone to build "easier" interpretable rule sets than others and to understand the reason why this happens; (*iii*) last but not least, we would like to investigate the computational aspect: so far, for some ontologies, a learning run may take even a week (on the resource at our disposal). Here, we would like to investigate both parallesization methods as well as to investigate about the impact, in terms of effectiveness, of efficient, logically sound, but not necessarily complete, reasoning algorithms.

A References related to learning w.r.t. ontologies

Find below an extensive list of references related to learning w.r.t. ontologies (the grouping is tentative).

Refinement operators: $\mathcal{EL}$-like [18, 59], general [60, 61, 62, 5].

Decision Trees/Random Forests: [39, 41, 98, 97, 99, 95, 106, 102, 104]

Kernel Methods: [8, 33, 35, 42]

FOIL-like: [34, 32, 45]

Boosting: [44]

Genetic Programming: [56]

Relational Learning: [94]

Naive Bayes: [77, 78, 81, 124]

Reinforcement Learning: [84]

Query Answering: [20, 22, 21, 24, 25, 26, 36, 38, 40, 82, 96, 103]

Clustering: [31, 37, 51, 50]

Regression: [43, 100]

Fuzzy: [53, 55, 72, 67, 68, 74, 75, 69, 70, 66, 114]

Others: [23, 27, 49, 52, 58, 63, 65, 64, 71, 79, 80, 93, 101, 105, 115, 116, 118, 120, 119, 117, 121]

Named Algorithms and Systems: $\mathcal{AL}$-LOG [65, 64], OCEL [60], YINYANG [52], DL-FOIL [34], $\mathcal{EL}$ TREE LEARNER [59], DL-LEARNER [57, 15, 16], CELOE [58], PARCEL [115], SPACEL [117], DL-FOCL [105], *Soft*FOIL [73, 67], FUZZYDL-LEARNER [70], PFOIL [114], BELNET^{+} [124]

References

References

[1] *Special issue on Fuzzy Clustering, Fuzzy Sets and Systems*, volume 389. Elsevier, 2020.

[2] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[3] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 364–369, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[4] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the 30th Annual German Conference on Advances in Artificial Intelligence (KI-07)*, number 4667 in Lecture Notes in Computer Science, pages 52–67, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Liviu Badea and Shan-Hwei Nienhuys-Cheng. A refinement operator for description logics. In *Inductive Logic Programming, 10th International Conference, ILP-00*, volume 1866 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2000.

[6] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., 1999.

[7] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Springer Verlag, 1981.

[8] Stephan Bloehdorn and York Sure. Kernel methods for mining instance data in ontologies. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *Lecture Notes in Computer Science*, pages 58–71. Springer Verlag, 2007.

[9] Fernando Bobillo, Marco Cerami, Francesc Esteva, Àngel García-Cerdaña, Rafael Peñaloza, and Umberto Straccia. Fuzzy description logics in the framework of mathematical fuzzy logic. In Carles Noguera Petr Cintula, Christian Fermüller, editor, *Handbook of Mathematical Fuzzy Logic, Volume 3*, volume 58 of *Studies in Logic, Mathematical Logic and Foundations*, chapter 16, pages 1105–1181. College Publications, 2015.

[10] Fernando Bobillo and Umberto Straccia. fuzzyDL: An expressive fuzzy description logic reasoner. In *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages 923–930. IEEE Computer Society, 2008.

[11] Fernando Bobillo and Umberto Straccia. Representing fuzzy ontologies in owl 2. In *Proceedings of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pages 2695–2700. IEEE Press, July 2010.

[12] Fernando Bobillo and Umberto Straccia. Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, 52:1073–1094, 2011.

[13] Fernando Bobillo and Umberto Straccia. The fuzzy ontology reasoner *fuzzyDL*. *Knowledge-Based Systems*, 95:12 – 34, 2016.

[14] Fernando Bobillo and Umberto Straccia. Reasoning within fuzzy owl 2 el revisited. *Fuzzy Sets and Systems*, 351:1–40, 2018.

[15] Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. Dl-learner - A framework for inductive learning on the semantic web. *Journal of Web Semantics*, 39:15–24, 2016.

[16] Lorenz Bühmann, Jens Lehmann, Patrick Westphal, and Simon Bin. Dl-learner structured machine learning on semantic web data. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 467–471. ACM, 2018.

[17] Marco Cerami and Umberto Straccia. On the (un)decidability of fuzzy description logics under lukasiewicz t-norm. *Information Sciences*, 227:1–21, 2013.

[18] Mahsa Chitsaz, Kewen Wang, Michael Blumenstein, and Guilin Qi. Concept learning for $\mathcal{EL}^{++}$; by refinement and reinforcement. In *Proceedings of the 12th Pacific Rim international conference on Trends in Artificial Intelligence*, PRICAI'12, pages 15–26, Berlin, Heidelberg, 2012. Springer-Verlag.

[19] Marcos E. Cintra, Maria Carolina Monard, and Heloisa de Arruda Camargo. On rule learning methods: A comparative analysis of classic and fuzzy approaches. In *Soft Computing: State of the Art Theory and Novel Applications*, volume 291, pages 89–104. Springer Verlag, 2013.

[20] Claudia d'Amato and Nicola Fanizzi. Lazy learning from terminological knowledge bases. In *Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings*, volume 4203 of *Lecture Notes in Computer Science*, pages 570–579. Springer Verlag, 2006.

[21] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Distance-based classification in OWL ontologies. In *Knowledge-Based Intelligent Information and Engineering Systems, 12th International Conference, KES-08*, volume 5178 of *Lecture Notes in Computer Science*, pages 656–661, 2008.

[22] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Query answering and ontology population: An inductive approach. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, volume 5021 of *Lecture Notes in Computer Science*, pages 288–302. Springer Verlag, 2008.

[23] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Inductive learning for the semantic web: What does it buy? *Semantic Web*, 1(1-2):53–59, 2010.

[24] Claudia d'Amato, Nicola Fanizzi, Bettina Fazzinga, Georg Gottlob, and Thomas Lukasiewicz. Combining semantic web search with the power of inductive reasoning. In *Scalable Uncertainty Management - 4th International Conference, SUM 2010, Toulouse, France, September 27-29, 2010. Proceedings*, volume 6379 of *Lecture Notes in Computer Science*, pages 137–150, 2010.

[25] Claudia d'Amato, Nicola Fanizzi, Bettina Fazzinga, Georg Gottlob, and Thomas Lukasiewicz. Ontology-based semantic search on the web and its combination with the power of inductive reasoning. *Annals of Mathematics and Artificial Intelligence*, 65(2-3):83–121, 2012.

[26] Claudia d'Amato, Nicola Fanizzi, Bettina Fazzinga, Georg Gottlob, and Thomas Lukasiewicz. Semantic web search and inductive reasoning. In *Uncertainty Reasoning for the Semantic Web II, International Workshops URSW 2008-2010 Held at ISWC and UniDL 2010 Held at FLoC, Revised Selected Papers*, volume 7123 of *Lecture Notes in Computer Science*, pages 237–261. Springer Verlag, 2013.

[27] Claudia d'Amato, Nicola Fanizzi, Marko Grobelnik, Agnieszka Lawrynowicz, and Vojtech Svátek. Inductive reasoning and machine learning for the semantic web. *Semantic Web*, 5(1):3–4, 2014.

[28] María José del Jesús, Frank Hoffmann, Luis Junco Navascués, and Luciano Sánchez. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3):296–308, 2004.

[29] Mario Drobics, Ulrich Bodenhofer, and Erich-Peter Klement. Fs-foil: an inductive learning method for extracting interpretable fuzzy descriptions. *International Journal of Approximate Reasoning*, 32(2-3):131–152, 2003.

[30] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[31] Floriana Esposito, Claudia d'Amato, and Nicola Fanizzi. Fuzzy clustering for semantic knowledge bases. *Fundamenta Informatica*, 99(2):187–205, 2010.

[32] Nicola Fanizzi. Concept induction in description logics using information-theoretic heuristics. *Int. J. Semantic Web Inf. Syst.*, 7(2):23–44, 2011.

[33] Nicola Fanizzi and Claudia d'Amato. A declarative kernel for $\mathcal{ALC}$ concept descriptions. In *Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings*, volume 4203 of *Lecture Notes in Computer Science*, pages 322–331. Springer Verlag, 2006.

[34] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL concept learning in description logics. In Filip Zelezný and Nada Lavrač, editors, *Inductive Logic Programming*, volume 5194 of *Lecture Notes in Computer Science*, pages 107–121. Springer, 2008.

[35] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Induction of classifiers through non-parametric methods for approximate classification and retrieval with ontologies. *Int. J. Semantic Computing*, 2(3):403–423, 2008.

[36] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Statistical learning for inductive query answering on OWL ontologies. In *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pages 195–212. Springer Verlag, 2008.

[37] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Fuzzy clustering for categorical spaces. In *Foundations of Intelligent Systems, 18th International Symposium, ISMIS-09*, volume 5722 of *Lecture Notes in Computer Science*, pages 161–170, 2009.

[38] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Reduce: A reduced coulomb energy network method for approximate classification. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 of *Lecture Notes in Computer Science*, pages 323–337. Springer, 2009.

[39] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Induction of concepts in web ontologies through terminological decision trees. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part I*, volume 6321 of *Lecture Notes in Computer Science*, pages 442–457. Springer Verlag, 2010.

[40] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Towards learning to rank in description logics. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 985–986. IOS Press, 2010.

[41] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Towards the induction of terminological decision trees. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1423–1427, New York, NY, USA, 2010. ACM.

[42] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Induction of robust classifiers for web ontologies through kernel machines. *J. Web Sem.*, 11:1–13, 2012.

[43] Nicola Fanizzi, Claudia d'Amato, Floriana Esposito, and Pasquale Minervini. Numeric prediction on OWL knowledge bases through terminological regression trees. *Int. J. Semantic Computing*, 6(4):429–446, 2012.

[44] Nicola Fanizzi, Giuseppe Rizzo, and Claudia d'Amato. Boosting DL concept learners. In *The Semantic Web - 16th International Conference, ESWC-19*, volume 11503 of *Lecture Notes in Computer Science*, pages 68–83, 2019.

[45] Nicola Fanizzi, Giuseppe Rizzo, Claudia d'Amato, and Floriana Esposito. Dlfoil: Class expression learning revisited. In *Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW-18*, volume 11313 of *Lecture Notes in Computer Science*, pages 98–113, 2018.

[46] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[47] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics 28*, 28:337–374, 2000.

[48] Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[49] Sebastian Hellmann, Jens Lehmann, and Sören Auer. Learning of OWL class descriptions on very large knowledge bases. *Int. J. Semantic Web Inf. Syst.*, 5(2):25–48, 2009.

[50] Ignacio Huitzil, Fernando Bobillo, Juan Gomez-Romero, and Umberto Straccia. Fudge: Fuzzy ontology building with consensuated fuzzy datatypes. *Fuzzy Sets and Systems*, xxx:xxx–xxx, 2020.

[51] Ignacio Huitzil, Umberto Straccia, Natalia Díaz-Rodríguez, and Fernando Bobillo. Datil: Learning fuzzy ontology datatypes. In *Proceedings of the 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2018), Part II*, volume 854 of *Communications in Computer and Information Science*, pages 100–112. Springer, June 2018.

[52] Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, 26(2):139–159, April 2007.

[53] Josué Iglesias and Jens Lehmann. Towards integrating fuzzy logic capabilities into an ontology-based inductive logic programming framework. In *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011)*, pages 1323–1328, 2011.

[54] George J. Klir and Bo Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[55] S. Konstantopoulos and A. Charalambidis. Formulating description logic learning as an inductive logic programming task. In *Proceedings of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pages 1–7. IEEE Press, 2010.

[56] Jens Lehmann. Hybrid learning of ontology classes. In *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, volume 4571 of *Lecture Notes in Computer Science*, pages 883–898. Springer Verlag, 2007.

[57] Jens Lehmann. DL-Learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 10:2639–2642, 2009.

[58] Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. Class expression learning for ontology engineering. *J. Web Sem.*, 9(1):71–81, 2011.

[59] Jens Lehmann and Christoph Haase. Ideal downward refinement in the *EL*\mathcal{EL} description logic. In *Inductive Logic Programming, 19th International Conference, ILP 2009, Leuven, Belgium, July 02-04, 2009. Revised Papers*, volume 5989 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2009.

[60] Jens Lehmann and Pascal Hitzler. A refinement operator based learning algorithm for the *ALC* description logic. In *Inductive Logic Programming, 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers*, volume 4894 of *Lecture Notes in Computer Science*, pages 147–160. Springer Verlag, 2007.

[61] Jens Lehmann and Pascal Hitzler. Foundations of Refinement Operators for Description Logics. In H. Blockeel, J. Ramon, J. W. Shavlik, and P. Tadepalli, editors, *Inductive Logic Programming*, volume 4894 of *Lecture Notes in Artificial Intelligence*, pages 161–174. Springer, 2008.

[62] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1-2):203–250, 2010.

[63] Jens Lehmann and Johanna Völker. *Perspectives on Ontology Learning*, volume 18 of *Studies on the Semantic Web*. IOS Press, 2014.

[64] Francesca A. Lisi and Floriana Esposito. Efficient evaluation of candidate hypotheses in al-log. In *Inductive Logic Programming, 14th International Conference, ILP 2004, Porto, Portugal, September 6-8, 2004, Proceedings*, volume 3194 of *Lecture Notes in Computer Science*, pages 216–233. Springer, 2004.

[65] Francesca A. Lisi and Donato Malerba. Ideal refinement of descriptions in al-log. In *Inductive Logic Programming: 13th International Conference, ILP-03*, volume 2835 of *Lecture Notes in Computer Science*, pages 215–232, 2003.

[66] Francesca A. Lisi and Corrado Mencar. A granular computing method for OWL ontologies. *Fundamenta Informatica*, 159(1-2):147–174, 2018.

[67] Francesca A. Lisi and Umberto Straccia. A logic-based computational method for the automated induction of fuzzy ontology axioms. *Fundamenta Informaticae*, 124(4):503–519, 2013.

[68] Francesca A. Lisi and Umberto Straccia. A system for learning GCI axioms in fuzzy description logics. In *Proceedings of the 26th International Workshop on Description Logics (DL-13)*, volume 1014 of *CEUR Workshop Proceedings*, pages 760–778. CEUR-WS.org, 2013.

[69] Francesca A. Lisi and Umberto Straccia. Can ilp deal with incomplete and vague structured knowledge? In Stephen H. Muggleton and Hiroaki Watanabe, editors, *Latest Advances in Inductive Logic Programming*, chapter 21, pages 199–206. World Scientific, 2014.

[70] Francesca A. Lisi and Umberto Straccia. Learning in description logics with fuzzy concrete domains. *Fundamenta Informaticae*, 140(3-4):373–391, 2015.

[71] Francesca Alessandra Lisi. Logics in machine learning and data mining: Achievements and open issues. In *Proceedings of the 34th Italian Conference on Computational Logic, Trieste, Italy, June 19-21, 2019.*, volume 2396 of *CEUR Workshop Proceedings*, pages 82–88. CEUR-WS.org, 2019.

[72] Francesca Alessandra Lisi and Umberto Straccia. An inductive logic programming approach to learning inclusion axioms in fuzzy description logics. In *26th Italian Conference on Computational Logic (CILC-11)*, volume 810, pages 57–71. CEUR Electronic Workshop Proceedings, 2011.

[73] Francesca Alessandra Lisi and Umberto Straccia. Towards learning fuzzy dl inclusion axioms. In *9th International Workshop on Fuzzy Logic and Applications (WILF-11)*, volume 6857 of *Lecture Notes in Computer Science*, pages 58–66, Berlin, 2011. Springer Verlag.

[74] Francesca Alessandra Lisi and Umberto Straccia. Dealing with incompleteness and vagueness in inductive logic programming. In *28th Italian Conference on Computational Logic (CILC-13)*, volume 1068, pages 179–193. CEUR Electronic Workshop Proceedings, 2013.

[75] Francesca Alessandra Lisi and Umberto Straccia. A foil-like method for learning under incompleteness and vagueness. In *23rd International Conference on Inductive Logic Programming*, volume 8812 of *Lecture Notes in Artificial Intelligence*, pages 123–139, Berlin, 2014. Springer Verlag. Revised Selected Papers.

[76] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6:291–308, 2008.

[77] Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi. Learning terminological naive bayesian classifiers under different assumptions on missing knowledge. In *Proceedings of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW-11*, volume 778 of *CEUR Workshop Proceedings*, pages 63–74. CEUR-WS.org, 2011.

[78] Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi. Learning probabilistic description logic concepts: Under different assumptions on missing knowledge. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 378–383, New York, NY, USA, 2012. ACM.

[79] Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi. Learning terminological bayesian classifiers - A comparison of alternative approaches to dealing with unknown concept-memberships. In *Proceedings of the 9th Italian Convention on Computational Logic, Rome, Italy, June 6-7, 2012*, volume 857 of *CEUR Workshop Proceedings*, pages 191–205, 2012.

[80] Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Transductive inference for class-membership propagation in web ontologies. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 457–471. Springer Verlag, 2013.

[81] Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Learning probabilistic description logic concepts under alternative assumptions on incompleteness. In *Uncertainty Reasoning for the Semantic Web III - ISWC International Workshops, URSW 2011-2013, Revised Selected Papers*, volume 8816 of *Lecture Notes in Computer Science*, pages 184–201, 2014.

[82] Pasquale Minervini, Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Rank prediction for semantically annotated resources. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 333–338, New York, NY, USA, 2013. ACM.

[83] Boris Motik and Riccardo Rosati. A faithful integration of description logics with logic programming. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 477–482, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[84] Matthias Nickles and Achim Rettinger. Interactive relational reinforcement learning of concept semantics. *Machine Learning*, 94(2):169–204, 2014.

[85] Richard Nock and Frank Nielsen. A real generalization of discrete AdaBoost. In *17th European Conference on Artificial Intelligence (ECAI-06)*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 509–515. IOS Press, 2006.

[86] Richard Nock and Frank Nielsen. A real generalization of discrete AdaBoost. *Artificial Intelligence Journal*, 171(1):25–41, 2007.

[87] José Otero and Luciano Sánchez. Induction of descriptive fuzzy classifiers with the logitboost algorithm. *Soft Computing*, 10(9):825–835, 2006.

[88] OWL 2 Web Ontology Language Document Overview. *https://www.w3.org/TR/owl2-overview/*. W3C, 2009.

[89] OWL 2 Web Ontology Language Profiles: OWL 2 EL. *https://www.w3.org/TR/owl2-profiles/#OWL_2_EL*. W3C, 2009.

[90] Ana M. Palacios, Luciano Sánchez, and Inés Couso. Using the AdaBoost algorithm for extracting fuzzy rules from low quality data: Some preliminary results. In *FUZZ-IEEE 2011, IEEE International Conference on Fuzzy Systems, Taipei, Taiwan, 27-30 June, 2011, Proceedings*, pages 1263–1270. IEEE, 2011.

[91] Luc De Raedt. *Logical and relational learning*. Cognitive Technologies. Springer, 2008.

[92] Luc De Raedt and Kristian Kersting. Statistical relational learning. In *Encyclopedia of Machine Learning and Data Mining*, pages 1177–1187. Springer, 2017.

[93] Achim Rettinger, Uta Lösch, Volker Tresp, Claudia d'Amato, and Nicola Fanizzi. Mining the semantic web - statistical learning for next generation knowledge bases. *Data Minining and Knowledge Discovery*, 24(3):613–662, 2012.

[94] Achim Rettinger, Matthias Nickles, and Volker Tresp. Statistical relational learning with formal ontologies. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II*, volume 5782 of *Lecture Notes in Computer Science*, pages 286–301. Springer Verlag, 2009.

[95] Giuseppe Rizzo, Claudia d'Amato, and Nicola Fanizzi. On the effectiveness of evidence-based terminological decision trees. In *Foundations of Intelligent Systems - 22nd International Symposium, ISMIS-15*, volume 9384 of *Lecture Notes in Computer Science*, pages 139–149, 2015.

[96] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Assertion prediction with ontologies through evidence combination. In *Uncertainty Reasoning for the Semantic Web II, International Workshops URSW 2008-2010 Held at ISWC and UniDL 2010 Held at FLoC, Revised Selected Papers*, volume 7123 of *Lecture Notes in Computer Science*, pages 282–299. Springer Verlag, 2013.

[97] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Tackling the class-imbalance learning problem in semantic web knowledge bases. In *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW-14*, volume 8876 of *Lecture Notes in Computer Science*, pages 453–468, 2014.

[98] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Towards evidence-based terminological decision trees. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part I*, volume 442 of *Communications in Computer and Information Science*, pages 36–45. Springer, 2014.

[99] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Inductive classification through evidence-based models and their ensembles. In *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC-15*, volume 9088 of *Lecture Notes in Computer Science*, pages 418–433, 2015.

[100] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Approximating numeric role fillers via predictive clustering trees for knowledge base enrichment in the web of data. In *Discovery Science - 19th International Conference, DS-16*, volume 9956 of *Lecture Notes in Computer Science*, pages 101–117, 2016.

[101] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Terminological cluster trees for disjointness axiom discovery. In *The Semantic Web - 14th International Conference, ESWC-17*, volume 10249 of *Lecture Notes in Computer Science*, pages 184–201, 2017.

[102] Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Tree-based models for inductive classification on the web of data. *Journal of Web Semantics*, 45:1–22, 2017.

[103] Giuseppe Rizzo, Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Prediction of class and property assertions on OWL ontologies through evidence combination. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 45:1–45:9, New York, NY, USA, 2011. ACM.

[104] Giuseppe Rizzo, Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Approximate classification with web ontologies through evidential terminological trees and forests. *International Journal of Approximate Reasoning*, 92:340–362, 2018.

[105] Giuseppe Rizzo, Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. A framework for tackling myopia in concept learning on the web of data. In *Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW-18*, volume 11313 of *Lecture Notes in Artificial Intelligence*, pages 338–354, 2018.

[106] Giuseppe Rizzo, Nicola Fanizzi, Jens Lehmann, and Lorenz Bühmann. Integrating new refinement operators in terminological decision trees learning. In *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW-16*, volume 10024 of *Lecture Notes in Computer Science*, pages 511–526, 2016.

[107] Luciano Sánchez and José Otero. Boosting fuzzy rules in classification problems under single-winner inference. *International Journal of Intelligent Systems*, 22(9):1021–1034, 2007.

[108] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[109] Mathieu Serrurier and Henri Prade. Improving expressivity of inductive logic programming by learning different kinds of fuzzy rules. *Soft Computing*, 11(5):459–466, 2007.

[110] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[111] D. Shibata, N. Inuzuka, S. Kato, T. Matsui, and H. Itoh. An induction algorithm based on fuzzy logic programming. In N. Zhong and L. Zhou, editors, *Methodologies for Knowledge Discovery and Data Mining, Third Pacific-Asia Conference, PAKDD-99, Beijing, China, April 26-28, 1999, Proceedings*, volume 1574 of *Lecture Notes in Computer Science*, pages 268–273. Springer, 1999.

[112] Umberto Straccia. Description logics with fuzzy concrete domains. In Fahiem Bachus and Tommi Jaakkola, editors, *21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland, 2005. AUAI Press.

[113] Umberto Straccia. *Foundations of Fuzzy Logic and Semantic Web Languages.* CRC Studies in Informatics Series. Chapman & Hall, 2013.

[114] Umberto Straccia and Matteo Mucci. pFOIL-DL: Learning (fuzzy) $\mathcal{EL}$ concept descriptions from crisp OWL data using a probabilistic ensemble estimation. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC-15)*, pages 345–352, Salamanca, Spain, 2015. ACM.

[115] An C. Tran, Jens Dietrich, Hans W. Guesgen, and Stephen Marsland. An approach to parallel class expression learning. In *Rules on the Web: Research and Applications - 6th International Symposium, RuleML 2012, Montpellier, France, August 27-29, 2012. Proceedings*, volume 7438, pages 302–316. Lecture Notes in Computer Science, 2012.

[116] An C. Tran, Jens Dietrich, Hans W. Guesgen, and Stephen Marsland. Two-way parallel class expression learning. In *Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012*, JMLR Proceedings, pages 443–458. JMLR.org, 2012.

[117] An C. Tran, Jens Dietrich, Hans W. Guesgen, and Stephen Marsland. Parallel symmetric class expression learning. *Journal of Machine Learning Research*, 18:64:1–64:34, 2017.

[118] An C. Tran, Hans W. Guesgen, Jens Dietrich, and Stephen Marsland. An approach to numeric refinement in description logic learning for learning activities duration in smart homes. In *Space, Time, and Ambient Intelligence, Papers from the 2013 AAAI Workshop, Bellevue, Washington, USA, July 14, 2013*, volume WS-13-14 of *AAAI Workshops*. AAAI, 2013.

[119] An Cong Tran. Application of description logic learning in abnormal behaviour detection in smart homes. In *The 2015 IEEE RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future, RIVF 2015, Can Tho, Vietnam, January 25-28, 2015*, pages 7–12. IEEE, 2015.

[120] Thanh-Luong Tran, Quang-Thuy Ha, Thi-Lan-Giao Hoang, Linh Anh Nguyen, and Hung Son Nguyen. Bisimulation-based concept learning in description logics. *Fundamenta Informatica*, 133(2-3):287–303, April 2014.

[121] Patrick Westphal, Lorenz Bühmann, Simon Bin, Hajira Jabeen, and Jens Lehmann. SML-bench - A benchmarking framework for structured machine learning. *Semantic Web*, 10(2):231–245, 2019.

[122] Hong yang Zhu, Yi Ding, Hong Gao, and Wei Liu. Fuzzy prediction in classification of AdaBoost algorithm. In *International Conference on Oriental Thinking and Fuzzy Logic*, volume 443 of *Advances in Intelligent Systems and Computing*, pages 129–136. Springer, 2016.

[123] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[124] Man Zhu, Zhiqiang Gao, Jeff Z. Pan, Yuting Zhao, Ying Xu, and Zhibin Quan. Tbox learning from incomplete data by inference in belnet+. *Knoweledge-Based Systems*, 75:30–40, 2015.