PN-OWL: A Two Stage Algorithm to Learn Fuzzy Concept Inclusions from OWL Ontologies

In *Fuzzy Sets and Systems*, vol. 490, nr. 109048, August 2024

Umberto Straccia

CNR-ISTI, Pisa, Italy

In Top-2% Scientists Ranking
Elsevier/Stanford University, 09/2024

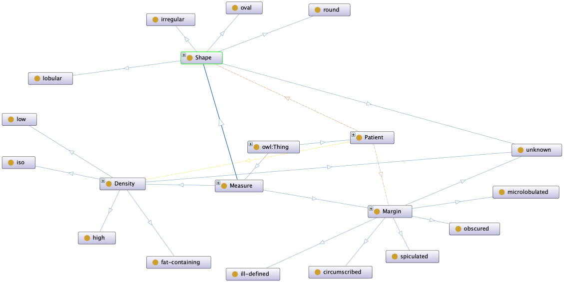
umberto.straccia@isti.cnr.it
www.umbertostraccia.it/cs

Consiglio Nazionale delle Ricerche
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
STARWARS
Funded by the European Union

Outline

Context & Problem Description (Informally)

Ontology-based learning and reasoning



Patient	hasDensity	hasShape	hasMargin	hasBiRads	hasAge	...
p0	low	lobular	spiculated	5	67	...
p10	high	irregular	spiculated	5	76	...
p102	-	irregular	ill-defined	4	58	...
p108	low	round	circumscribed	4	57	...
p109	-	irregular	ill-defined	5	33	...
p110	low	irregular	ill-defined	4	45	...
p111	low	irregular	ill-defined	5	71	...
...	...	...	...	...	...	...

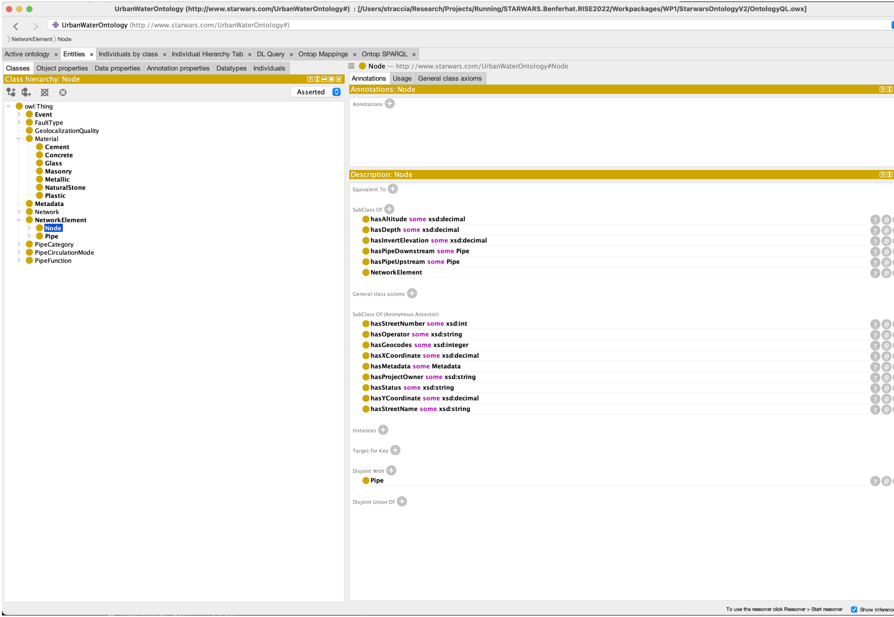
- Input: Mammography patient data & Mammography ontology
- Output: Learned Rules to decide “Does a patient have breast cancer?”

- First, we need to know what is the
  - input data language
  - output rule language
- Then, we may address how to learn the output from the input

Ontology-based Learning

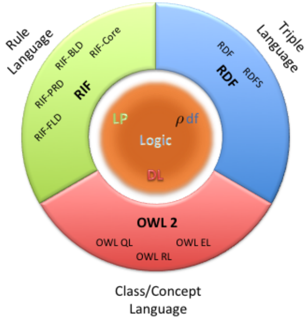
- Data is represented by means of an ontology
- An ontology is a description of a set of conceptual entities of a domain that shows their properties and the relations between them
- It ensures a common understanding of information and makes explicit domain assumptions
  - Allows organizations to make better sense of their data
- Ontologies do not only represent sharable and reusable knowledge, but can also used to infer new knowledge about a domain
- To enable such a representation, we need to formally specify components such as individuals, classes, attributes and relations as well as restrictions, rules and axioms

Urban Water Network Ontology



The Semantic Web Family of Languages

- Semantic Web family of languages widely used to specify ontologies, viz. structured data
- Wide variety of languages
  - RDFS: *Triple language*, -*Resource Description Framework*
    - The logical counterpart is ρdf
  - RIF: *Rule language*, -*Rule Interchange Format*,
    - Relate to the Logic Programming (LP) paradigm
  - OWL 2: *Conceptual language*, -*Ontology Web Language*
    - Relate to Description Logics (DLs)

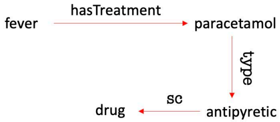


Resource Description Framework Schema (RDFS)

- RDFS: W3C standard and popular logic for KR
- Statements
  - Triples of the form (s, p, o)
  - Informally, binary predicate $p(s, o)$

$$(\texttt{fever}, \texttt{hasTreatment}, \texttt{paracetamol})$$

  - Special predicates: typing and specialisations, etc.
    $(\texttt{paracetamol}, \text{type}, \texttt{antipyretic})$
    $(\texttt{antipyretic}, \text{sc}, \texttt{drug})$



- *Knowledge Graphs* may be seen as a special case

ρdf

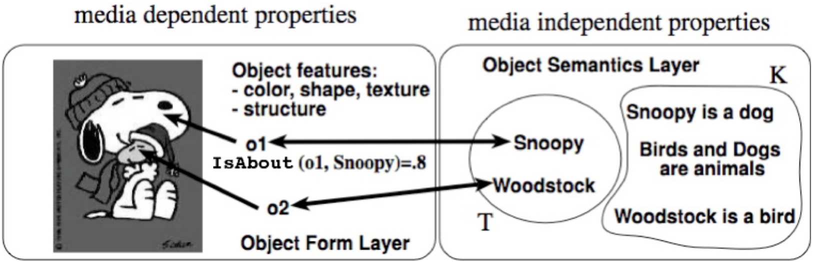
- The logic ρdf
  - A minimal, but significant RDFS fragment
  - Covers all essential features of RDFS
- ρdf: defined on subset of the RDFS vocabulary:

$$\rho\text{df} = \{\text{sp}, \text{sc}, \text{type}, \text{dom}, \text{range}\}$$

Informally,

- (p, sp, q)
  - p is a sub property of property q
- (c, sc, d)
  - c is a sub class of class d
- (a, type, b)
  - a is of type b
- (p, dom, c)
  - domain of property p is c
- $(p, \textit{range}, c)$
  - range of property p is c

Example



$$G = \left\{ \begin{array}{ll} (\texttt{o1}, \texttt{IsAbout}, \texttt{snoopy}) & (\texttt{o2}, \texttt{IsAbout}, \texttt{woodstock}) \\ (\texttt{snoopy}, \text{type}, \texttt{dog}) & (\texttt{woodstock}, \text{type}, \texttt{bird}) \\ (\texttt{dog}, \text{sc}, \texttt{animal}) & (\texttt{bird}, \text{sc}, \texttt{animal}) \end{array} \right\}$$

- Computational complexity of query answering:
  - $O(n^3)$, n number of triples

Rule Languages, viz. Logic Programs (for ease, Datalog)

- Predicates are n-ary
- Terms are variables or constants
- Facts ground atoms
  For instance,
  $$has\_parent(mary, tom)$$
- Rules are of the form
  $$P(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$$
  where
  - $\varphi(\mathbf{x}, \mathbf{y})$ is a formula built from atoms of the form $B(\mathbf{z})$ and connectors $\wedge, \vee, 0, 1$
  - $\mathbf{z}_i$ is a tuple of literals, or variables in $\mathbf{x}, \mathbf{y}$
- For instance,
  $$has\_father(x, y) \quad \leftarrow \quad has\_parent(x, y) \wedge Male(y)$$

Remark

Note that

$$has\_father(x, y) \quad \leftarrow \quad has\_parent(x, y), Male(y)$$

is the same as repplacing "," with $\wedge$

- Extensional database (EDB): set of facts
- Intentional database (IDB): set of rules
- Logic Program $\mathcal{P}$:
  - $\mathcal{P} = \mathit{EDB} \cup \mathit{IDB}$
  - No predicate symbol in *EDB* occurs in the head of a rule in *IDB*
    - The principle is that we do not allow that *IDB* may redefine the extension of predicates in *EDB*
- *EDB* is usually, stored into a relational database
- Computational complexity of query answering
  - Polynomial w.r.t. $|\mathit{EDB}|$
  - EXPTIME w.r.t. $|\mathcal{P}|$

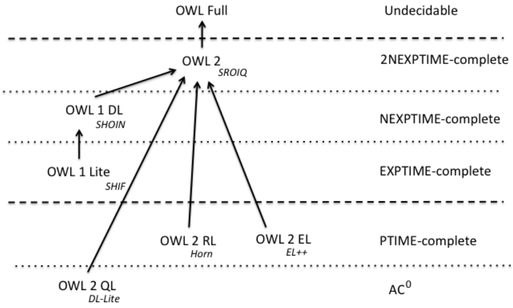
The Web Ontology Language OWL 2

- OWL 2 is a family of the object oriented languages

`class`	Person `partial` Human
	`restriction` (hasName `someValuesFrom String`)
	`restriction` (hasBirthPlace `someValuesFrom` Geoplace)

"The class Person *is a subclass of class* Human *and has two attributes:* hasName *having a string as value, and* hasBirthPlace *whose value is an instance of the class* Geoplace*".*

- Description Logics are the logics that stand behind OWL 2
- OWL languages differntiate in syntax and computational complexity of reasoning problems



OWL 2 Profiles

OWL 2 EL
- Useful for large size of properties and/or classes
- The EL acronym refers to the $\mathcal{EL}$ family of DLs

OWL 2 QL
- Useful for very large volumes of instance data
- Conjunctive query answering via query rewriting and SQL
- OWL 2 QL relates to the DL family *DL-Lite*

OWL 2 RL
- Useful for scalable reasoning without sacrificing too much expressive power
- OWL 2 RL maps to Datalog (an LP language)
- Computational complexity: same as for Datalog, polynomial in size of the data, EXPTIME w.r.t. size of knowledge base

Description Logics (DLs)

The logics behind OWL 2 and its profiles, `http://dl.kr.org/`

- Concept/Class: are unary predicates
- Role or attribute: binary predicates
- Taxonomy: Concept and role hierarchies can be expressed
- Individual: constants
- Operators: to build complex classes out from class names

- Basic ingredients:
  - a:C, called concept assertion, meaning that individual a is an instance of concept/class C

$$\text{a:Person} \sqcap \exists \text{hasChild.Femal}$$

  - (a, b):R, called role assertion, meaning that the pair of individuals $\langle a, b \rangle$ is an instance of the property/role R

$$(\text{tom}, \text{mary})\text{:hasChild}$$

  - $C \sqsubseteq D$, called General Concept Inclusion (GCI), meaning that the class C is a subclass of class D

$$\text{Father} \sqsubseteq \mathit{Male} \sqcap \exists \text{hasChild.Person}$$

Example (Toy Example)

$$\begin{aligned} Sex &= Male \sqcup Female \\ Male \sqcap Female &\sqsubseteq \bot \\ Person &\sqsubseteq Human \sqcap \exists hasSex.Sex \\ MalePerson &= Person \sqcap \exists hasSex.Male \\ & \quad functional(hasSex) \end{aligned}$$

$$umberto{:}Person \sqcap \exists hasSex.\neg Female$$

$$KB \models umberto{:}MalePerson$$

The DL Family

- A given DL is defined by set of concept and role forming operators
- Basic language: $\mathcal{ALC}$ ($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement)

Syntax				Semantics	Example
C, D	$\rightarrow$	$\top$	|	$\top(x)$	
		$\bot$	|	$\bot(x)$	
		A	|	$A(x)$	$Human$
		$C \sqcap D$	|	$C(x) \wedge D(x)$	$Human \sqcap Male$
		$C \sqcup D$	|	$C(x) \vee D(x)$	$Nice \sqcup Rich$
		$\neg C$	|	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	|	$\exists y.R(x, y) \wedge C(y)$	$\exists has\_child.Blond$
		$\forall R.C$		$\forall y.R(x, y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$				$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$				$C(a)$	$John{:}Happy\_Father$

DL Semantics

- Semantics is given in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where
  - $\Delta^{\mathcal{I}}$ is the domain (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an interpretation function that maps:
    - Concept (class) name A into a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
    - Role (property) name R into a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
    - Individual name a into an element of $\Delta^{\mathcal{I}}$
  - Interpretation function $\cdot^{\mathcal{I}}$ is extended to concept expressions:

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1{}^{\mathcal{I}} \cap C_2{}^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1{}^{\mathcal{I}} \cup C_2{}^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y \rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}
\end{aligned}
$$

Note on DL Naming

$\mathcal{AL}$: $C, D \longrightarrow \top \mid \bot \mid A \mid C \sqcap D \mid \neg A \mid \exists R.\top \mid \forall R.C$

$\mathcal{C}$: Concept negation, $\neg C$. Thus, $\mathcal{ALC} = \mathcal{AL} + \mathcal{C}$

$\mathcal{S}$: Used for $\mathcal{ALC}$ with transitive roles $\mathcal{R}_+$

$\mathcal{U}$: Concept disjunction, $C_1 \sqcup C_2$

$\mathcal{E}$: Existential quantification, $\exists R.C$

$\mathcal{H}$: Role inclusion axioms, $R_1 \sqsubseteq R_2$, e.g. $is\_component\_of \sqsubseteq is\_part\_of$

$\mathcal{N}$: Number restrictions, $(\geq n\ R)$ and $(\leq n\ R)$, e.g. $(\geq 3\ has\_Child)$ (has at least 3 children)

$\mathcal{Q}$: Qualified number restrictions, $(\geq n\ R.C)$ and $(\leq n\ R.C)$, e.g. $(\leq 2\ has\_Child.Adult)$ (has at most 2 adult children)

$\mathcal{O}$: Nominals (singleton class), $\{a\}$, e.g. $\exists has\_child.\{mary\}$.
Note: $a{:}C$ equiv to $\{a\} \sqsubseteq C$ and $(a, b){:}R$ equiv to $\{a\} \sqsubseteq \exists R.\{b\}$

$\mathcal{I}$: Inverse role, R^-, e.g. $isPartOf = hasPart^-$

$\mathcal{F}$: Functional role, f, e.g. $functional(hasAge)$

$\mathcal{R}_+$: transitive role, e.g. $transitive(isPartOf)$

For instance,

$$\mathcal{SHIF} = \mathcal{S} + \mathcal{H} + \mathcal{I} + \mathcal{F} = \mathcal{ALCR_+HIF} \quad \text{OWL-Lite}$$
$$\mathcal{SHOIN} = \mathcal{S} + \mathcal{H} + \mathcal{O} + \mathcal{I} + \mathcal{N} = \mathcal{ALCR_+HOIN} \quad \text{OWL-DL}$$
$$\mathcal{SROIQ} = \mathcal{S} + \mathcal{R} + \mathcal{O} + \mathcal{I} + \mathcal{Q} = \mathcal{ALCR_+ROIN} \quad \text{OWL 2}$$

Semantics of Additional Constructs

$\mathcal{H}$: Role inclusion axioms, $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff ${R_1}^{\mathcal{I}} \subseteq {R_1}^{\mathcal{I}}$

$\mathcal{N}$: Number restrictions,
$(\geq n\, R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\}| \geq n\}$,
$(\leq n\, R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\}| \leq n\}$

$\mathcal{Q}$: Qualified number restrictions,
$(\geq n\, R.C)^{\mathcal{I}} = \{x \in |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \geq n\}$,
$(\leq n\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \leq n\}$

$\mathcal{O}$: Nominals (singleton class), $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$

$\mathcal{I}$: Inverse role, $(R^{-})^{\mathcal{I}} = \{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$

$\mathcal{F}$: Functional role, $I \models fun(f)$ iff $\forall z \forall y \forall z$ if $\langle x, y\rangle \in f^{\mathcal{I}}$ and $\langle x, z\rangle \in f^{\mathcal{I}}$ the $y = z$

$\mathcal{R}_+$: transitive role,
$(R_+)^{\mathcal{I}} = \{\langle x, y\rangle \mid \exists z$ such that $\langle x, z\rangle \in R^{\mathcal{I}} \wedge \langle z, y\rangle \in R^{\mathcal{I}}\}$

Basics on Concrete Domains

- Concrete domains: reals, integers, strings, . . .
  $(tim, 14)$*:hasAge*
  $(sf,$ "*SoftComputing*"$)$*:hasAcronym*
  $(source1,$ "*ComputerScience*"$)$*:isAbout*
  $(service2,$ "*InformationRetrievalTool*"$)$*:Matches*
  $Minor = Person \sqcap \exists hasAge. \leq_{18}$

- Semantics: a clean separation between "object" classes and concrete domains
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete domain predicates d with a predefined arity n and fixed interpretation $d^D \subseteq \Delta_D^n$
  - Concrete properties: $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_D$

- Notation: (D). E.g., $\mathcal{ALC}(D)$ is $\mathcal{ALC}$ + concrete domains

DL Knowledge Base

- A DL Knowledge Base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where
  - $\mathcal{T}$ is a TBox
    - containing general inclusion axioms of the form $C \sqsubseteq D$,
    - concept definitions of the form $A = C$
    - primitive concept definitions of the form $A \sqsubseteq C$
    - role inclusions of the form $R \sqsubseteq P$
    - role equivalence of the form $R = P$
  - $\mathcal{A}$ is a ABox
    - containing assertions of the form $a{:}C$
    - containing assertions of the form $(a, b){:}R$
- An interpretation $\mathcal{I}$ is a model of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$ iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$, where
  - $\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ is a model of $\mathcal{T}$) iff $\mathcal{I}$ is a model of each element in $\mathcal{T}$
  - $\mathcal{I} \models \mathcal{A}$ ($\mathcal{I}$ is a model of $\mathcal{A}$) iff $\mathcal{I}$ is a model of each element in $\mathcal{A}$

Syntax and semantics of the DL $\mathcal{SROIQ}(\mathbf{D})$ (OWL 2)

Concepts	**Syntax (C)**	**FOL Reading of $C(x)$**
(C1)	A	$A(x)$
(C2)	$\top$	1
(C3)	$\bot$	0
(C4)	$C \sqcap D$	$C(x) \wedge D(x)$
(C5)	$C \sqcup D$	$C(x) \vee D(x)$
(C6)	$\neg C$	$\neg C(x)$
(C7)	$\forall R.C$	$\forall y.R(x,y) \rightarrow C(y)$
(C8)	$\exists R.C$	$\exists y.R(x,y) \wedge C(y)$
(C9)	$\forall T.\mathbf{d}$	$\forall v.T(x,v) \rightarrow \mathbf{d}(v)$
(C10)	$\exists T.\mathbf{d}$	$\exists v.T(x,v) \wedge \mathbf{d}(v)$
(C11)	$\{a\}$	$x = a$
(C12)	$(\geq m\ S.C)$	$\exists y_1 \ldots \exists y_m. \bigwedge_{i=1}^{m}(S(x,y_i) \wedge C(y_i)) \wedge \bigwedge_{1 \leq j < k \leq m} y_j \neq y_k$
(C13)	$(\leq m\ S.C)$	$\forall y_1 \ldots \forall y_{m+1}. \bigwedge_{i=1}^{m}(S(x,y_i) \wedge C(y_i)) \rightarrow \bigvee_{1 \leq j < k \leq m} y_j = y_k$
(C14)	$(\geq m\ T.\mathbf{d})$	$\exists v_1 \ldots \exists v_m. \bigwedge_{i=1}^{m}(T(x,v_i) \wedge \mathbf{d}(v_i)) \wedge \bigwedge_{1 \leq j < k \leq m} v_j \neq v_k$
(C15)	$(\leq m\ T.\mathbf{d})$	$\forall v_1 \ldots \forall v_{m+1}. \bigwedge_{i=1}^{m}(T(x,v_i) \wedge \mathbf{d}(v_i)) \rightarrow \bigvee_{1 \leq j < k \leq m} v_j = v_k$
(C16)	$\exists S.\text{Self}$	$S(x,x)$
Roles	**Syntax (R)**	**Semantics of $R(x,y)$**
(R1)	R	$R(x,y)$
(R2)	R^-	$R(y,x)$
(R3)	U	1

Axiom	Syntax (E)	Semantics ($\mathcal{I}$ satisfies E if ...)
(A1)	a:C	$C(a)$
(A2)	(a, b):R	$R(a, b)$
(A3)	(a, b):$\neg R$	$\neg R(a, b)$
(A4)	(a, v):T	$T(a, v)$
(A5)	(a, v):$\neg T$	$\neg T(a, v)$
(A6)	$C \sqsubseteq D$	$\forall x.C(x) \rightarrow D(x)$
(A7)	$R_1 \ldots R_n \sqsubseteq R$	$\forall x_1 \forall x_{n+1} \exists x_2 \ldots$ $\exists x_n.(R_1(x_1, x_2) \wedge \ldots \wedge R_n(x_n, x_{n+1})) \rightarrow R(x_1, x_{n+1})$
(A8)	$T_1 \sqsubseteq T_2$	$\forall x \forall v.T_1(x, v) \rightarrow T_2(x, v)$
(A9)	trans(R)	$\forall x \forall y \forall z.R(x, z) \wedge R(z, y) \rightarrow R(x, y)$
(A10)	disj(S_1, S_2)	$\forall x \forall y.S_1(x, y) \wedge S_2(x, y) = 0$
(A11)	disj(T_1, T_2)	$\forall x \forall v.T_1(x, v) \wedge T_2(x, v) = 0$
(A12)	ref(R)	$\forall x.R(x, x)$
(A13)	irr(S)	$\forall x.\neg S(x, x)$
(A14)	sym(R)	$\forall x \forall y.R(x, y) = R(y, x)$
(A15)	asy(S)	$\forall x \forall y, S(x, y) \rightarrow \neg S(y, x)$

OWL 2 as Description Logic (excerpt)

Concept/Class constructors:

Abstract Syntax	DL Syntax	Example
Descriptions (C)		
A (URI reference)	A	Conference
owl:Thing	$\top$	
owl:Nothing	$\bot$	
intersectionOf(C_1 C_2 ...)	$C_1 \sqcap C_2$	Reference $\sqcap$ Journal
unionOf(C_1 C_2 ...)	$C_1 \sqcup C_2$	Organization $\sqcup$ Institution
complementOf(C)	$\neg C$	$\neg$ MasterThesis
oneOf(o_1 ...)	$\{o_1, \ldots\}$	{"WISE","ISWC",...}
restriction(R someValuesFrom(C))	$\exists R.C$	$\exists$parts.InCollection
restriction(R allValuesFrom(C))	$\forall R.C$	$\forall$date.Date
restriction(R hasValue(o))	$\exists R.\{o\}$	$\exists$date.{2005}
restriction(R minCardinality(n))	$(\geq n\ R)$	($\geqslant$ 1 location)
restriction(R maxCardinality(n))	$(\leq n\ R)$	($\leqslant$ 1 publisher)
restriction(U someValuesFrom(D))	$\exists U.D$	$\exists$issue.integer
restriction(U allValuesFrom(D))	$\forall U.D$	$\forall$name.string
restriction(U hasValue(v))	$\exists U.=_v\}$	$\exists$series.$=_{\text{"LNCS"}}$
restriction(U minCardinality(n))	$(\geq n\ U)$	($\geqslant$ 1 title)
restriction(U maxCardinality(n))	$(\leq n\ U)$	($\leqslant$ 1 author)

Note: *R* is an abstract role, while *U* is a concrete property of arity two.

Axioms:

Abstract Syntax	DL Syntax	Example
Axioms		
`Class`(A `partial` $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$Human \sqsubseteq Animal \sqcap Biped$
`Class`(A `complete` $C_1 \ldots C_n$)	$A = C_1 \sqcap \ldots \sqcap C_n$	$Man = Human \sqcap Male$
`EnumeratedClass`($A\ o_1 \ldots o_n$)	$A = \{o_1\} \sqcup \ldots \sqcup \{o_n\}$	$RGB = \{r\} \sqcup \{g\} \sqcup \{b\}$
`SubClassOf`($C_1\ C_2$)	$C_1 \sqsubseteq C_2$	
`EquivalentClasses`($C_1 \ldots C_n$)	$C_1 = \ldots = C_n$	
`DisjointClasses`($C_1 \ldots C_n$)	$C_i \sqcap C_j = \bot, i \neq j$	$Male \sqcap Female \sqsubseteq \bot$
`ObjectProperty`(R `super` (R_1)... `super` (R_n)	$R \sqsubseteq R_i$	$HasDaughter \sqsubseteq hasChild$
`domain`(C_1)...`domain`(C_n)	$(\geq 1\ R) \sqsubseteq C_i$	$(\geq 1\ hasChild) \sqsubseteq Human$
`range`(C_1)...`range`(C_n)	$\top \sqsubseteq \forall R.C_i$	$\top \sqsubseteq \forall hasChild.Human$
[`inverseof`(P)]	$R = P^-$	$hasChild = hasParent^-$
[`symmetric`]	$R \sqsubseteq R^-$	$similar = similar^-$
[`functional`]	$\top \sqsubseteq (\leq 1\ R)$	$\top \sqsubseteq (\leq 1\ hasMother)$
[`Inversefunctional`]	$\top \sqsubseteq (\leq 1\ R^-)$	
[`Transitive`])	$Tr(R)$	$Tr(ancestor)$
`SubPropertyOf`($R_1\ R_2$)	$R_1 \sqsubseteq R_2$	
`EquivalentProperties`($R_1 \ldots R_n$)	$R_1 = \ldots = R_n$	$cost = price$
`AnnotationProperty`(S)		

Abstract Syntax	DL Syntax	Example
DatatypeProperty(U super (U_1)... super (U_n)	$U \sqsubseteq U_i$	
domain(C_1)...domain(C_n)	$(\geq 1\ U) \sqsubseteq C_i$	$(\geq 1\ hasAge) \sqsubseteq Human$
range(D_1)...range(D_n)	$\top \sqsubseteq \forall U.D_i$	$\top \sqsubseteq \forall hasAge.posInteger$
[functional])	$\top \sqsubseteq (\leq 1\ U)$	$\top \sqsubseteq (\leq 1\ hasAge)$
SubPropertyOf($U_1 U_2$)	$U_1 \sqsubseteq U_2$	$hasName \sqsubseteq hasFirstName$
EquivalentProperties($U_1 \ldots U_n$)	$U_1 = \ldots = U_n$	
Individuals		
Individual(o type (C_1)... type (C_n))	$o{:}C_i$	$tim{:}Human$
value($R_1 o_1$)...value($R_n o_n$)	$(o, o_i){:}R_i$	$(tim, mary){:}hasChild$
value($U_1 v_1$)...value($U_n v_n$)	$(o, v_1){:}U_i$	$(tim, 14){:}hasAge$
SameIndividual($o_1 \ldots o_n$)	$o_1 = \ldots = o_n$	$president\_Bush = G.W.Bush$
DifferentIndividuals($o_1 \ldots o_n$)	$o_i \neq o_j, i \neq j$	$john \neq peter$
Symbols		
Object Property R (URI reference)	R	hasChild
Datatype Property U (URI reference)	U	hasAge
Individual o (URI reference)	U	tim
Data Value v (RDF literal)	U	"International Conference on Semantic W

Basic Inference Problems (Formally)

Consistency: Check if knowledge is meaningful

- Is $\mathcal{K}$ satisfiability? $\mapsto$ Is there some model $\mathcal{I}$ of $\mathcal{K}$?
- Is C satisfiability? $\mapsto$ $C^{\mathcal{I}} \neq \emptyset$ for some some model $\mathcal{I}$ of $\mathcal{K}$?

Subsumption: structure knowledge, compute taxonomy

- $\mathcal{K} \models C \sqsubseteq D$? $\mapsto$ Is it true that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Equivalence: check if two classes denote same set of instances

- $\mathcal{K} \models C = D$? $\mapsto$ Is it true that $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Instantiation: check if individual a instance of class C

- $\mathcal{K} \models a{:}C$? $\mapsto$ Is it true that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Retrieval: retrieve set of individuals that are instances of calss C

- Compute the set $\{a \mid \mathcal{K} \models a{:}C\}$

Learning from OWL ontologies

- Input:
  - OWL ontology / CSV data
  - Classes, Object properties, real-valued data properties
  - target class T
    - the instances of T are the positive examples
    - all other individuals are the non-positive examples (negative examples + unlabelled individuals)

- Output: sufficient conditions, in terms of Fuzzy OWL EL Inclusion axioms , of being an individual instance of target class T
  - Roughly, set of Inclusion axioms of the form

$$\langle C \sqsubseteq T, n \rangle \quad n \in (0, 1] \text{ confidence/precision}$$

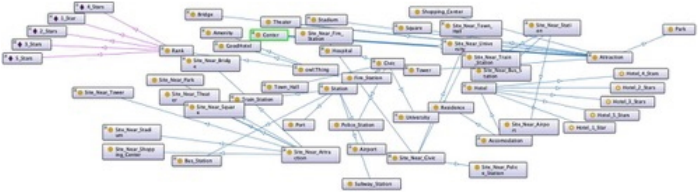
"If x instance of class C to degree m then it is a T to degree $m \cdot n$"

We have developed the Fuzzy DL-Learner: it supports various learning algorithms

- Fuzzy FOIL, pFOIL, Fuzzy Real AdaBoost, PN-OWL (new)
- order in increasing effectiveness

Examples

- Hotel ontology:



  - Target class *T*: `Good Hotel`, from TripAdvisor ratings

Learned: "An expensive B&B offering a cradle and Wi-Fi is a good hotel".

```
Bed_and_Breakfast and(hasAmenity some Cradle)and(hasAmenity some WI−FI)
    and(hasPrice some hasPrice_high)SubClassOf GoodHotel
```

- Mammography:

  Learned: "Mammography region of old woman, whose density mass is high, whose margin is spiculated, and whose shape is irregular is a tumor".

```
(hasAge some hasAgeHigh) and (hasDensity some hasDensityHigh) and (
    hasMargin some spiculated) and (hasShape some irregular)
    SubClassOf Tumor
```

In red: fuzzy classes

Formal Description

Given:

- a satisfiable crisp KB $\mathcal{K}$ and its individuals I_{KB};
- a *target concept name* T with associated unknown classification function $f_T : \mathbf{I} \rightarrow \{0,1\}$, where for each $a \in \mathbf{I}$, the possible values (*labels*) correspond, respectively, to $\mathcal{K} \not\models a{:}T$ (a is a *negative* example of T) and $\mathcal{K} \models a{:}T$ (a is a *positive* example of T);
- a *hypothesis space* of classifiers $\mathcal{H} = \{h_T : \mathsf{I}_{KB} \rightarrow \{0,1\}\}$;
- a *training set* $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^- \subset \mathsf{I}_{KB}$ of individuals, where

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

  on which f_T is known. We write $\mathcal{E}(a) = 1$ if a is a positive example, $\mathcal{E}(a) = 0$ if a is a negative example and assume that there is at least one positive example in $\mathcal{E}$;
- a *test set* $\bar{\mathcal{E}} \subset \mathsf{I}_{KB}$ of individuals with $\mathcal{E} \cap \bar{\mathcal{E}} = \emptyset$. We assume that $\bar{\mathcal{E}}$ contains at least one positive example. The conditions and notation used for the test set mirror those of the training set, with $\mathcal{E}$ replaced by $\bar{\mathcal{E}}$.

Learn:
a classifier $\bar{h} \in \mathcal{H}$ that approximates best the classification function according to the principle of *Empirical Risk Minimisation* (ERM) on $\mathcal{E}$: i.e. choose a classifier $\bar{h}$ such that:

$$
\bar{h} = \operatorname*{argmin}_{h \in \mathcal{H}} R(h, \mathcal{E})
$$

$$
R(h, \mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{a \in \mathcal{E}} L(h(a), \mathcal{E}(a)) ,
$$

with L being a *loss/error function* $L : \{0,1\}^2 \rightarrow \mathbb{R}$ such that $L(\hat{l}, l)$ measures how different the prediction $\hat{l}$ of a hypothesis is from the true outcome l: we use 0-1 *loss function*

$$
L(\hat{l}, l) = \begin{cases} 1 & \text{if } \hat{l} \neq l \\ 0 & \text{otherwise} \end{cases} ,
$$

and, thus, we try to minimise the number of misclassified training examples.

Evaluation Measures

Effectiveness of learnt classifier $\bar{h}$: assessed by means of effectiveness measures on the test set $\bar{\mathcal{E}}$.

Covered examples: individuals predicted being instance of T

True Positives: denoted TP, number of covered examples that are positive

False Positives: denoted FP, number of covered examples that that are not positive

Precision: denoted P, the fraction of true positives w.r.t. the covered examples ($TP + FP$)

$$P = \frac{TP}{TP + FP}$$

Recall: denoted R, fraction of true positives w.r.t. all positives (all instances of T)

$$R = \frac{TP}{|T|}$$

$F1$-score: denoted $F1$,

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

Accuracy: denoted Acc,

$$Acc = \frac{TP + TN}{|\mathsf{I}_{KB}|}$$

Misclassification Rate: denoted MR,

$$MR = \frac{FP + FN}{|\mathsf{I}_{KB}|} = 1 - Acc$$

Hypothesis Language

A hypothesis $h_T \in \mathcal{H}$ is a set of fuzzy GCIs/rules of the form

$$h_P = \{\langle C_1 \sqsubseteq P, \alpha_1 \rangle, \ldots, \langle C_h \sqsubseteq P, \alpha_h \rangle\} \quad (1)$$
$$h_N = \{\langle D_1 \sqsubseteq N, \beta_1 \rangle, \ldots, \langle D_k \sqsubseteq N, \beta_k \rangle\} \quad (2)$$

together with a decision operator of the form ($a \in \mathsf{I}_{KB}$)

$$h_T(a) = @(p_a, n_a) \ , \quad (3)$$

where

1. P and N are new class names not occurring in KB;
2. α_i, β_j are the inclusion degrees of the relative rules;
3. h_P is the set of P-rules as per Eq. 1;
4. h_N is the set of N-rules as per Eq. 2;
5. $p_a = bed(KB \cup h_P, a{:}P)$;
6. $n_a = bed(KB \cup h_N, a{:}N)$;
7. $@$ is an aggregation operator; and

8. each C_i, D_j is a fuzzy $\mathcal{EL}(D)$ concept expression defined as (b is a Boolean value)

$$
\begin{array}{lcl}
C & \longrightarrow & \top \mid A \mid \exists r.C \mid \exists s.\mathbf{d} \mid C_1 \sqcap C_2 \\
\mathbf{d} & \rightarrow & ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid \; =_b \; .
\end{array} \tag{4}
$$

with semantics ($\mathcal{I}$ is now a function into $[0, 1]$)

$$
\begin{array}{rcl}
\top^{\mathcal{I}}(x) & = & 1 \\
\bot^{\mathcal{I}}(x) & = & 0 \\
(C \sqcap D)^{\mathcal{I}}(x) & = & \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\} \\
(\exists r.C)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathcal{I}}} \{\min\{r^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y)\}\} \\
(\exists s.\mathbf{d})^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{D}} \{\min\{s^{\mathcal{I}}(x,y), \mathbf{d}^{D}(y)\}\} \; .
\end{array}
$$

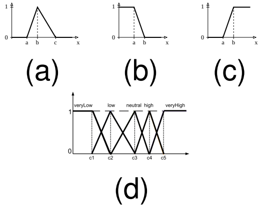


Figure: (a) triangular function $tri(a, b, c)$, (b) left shoulder function $ls(a, b)$, (c) right shoulder function $rs(a, b)$, and (d) fuzzy sets over centroids, learned from data via c-means clustering algorithm.

- Fuzzy axiom: $\langle \gamma, n \rangle$, where $\gamma \in \{C \sqsubseteq D, a{:}A, (a,b){:}r\}$
  - "γ is true to degree $\geq m$"
- $\mathcal{I}$ satisfies $\langle a{:}C, n \rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq n$
- $\mathcal{I}$ satisfies $\langle (a,b){:}r, n \rangle$ if $r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq n$
- $\mathcal{I}$ satisfies $\langle C \sqsubseteq D, n \rangle$ if for all $x \in \Delta^{D}$, $D^{\mathcal{I}}(x) \geq C^{\mathcal{I}}(x) \cdot n$
- $\mathcal{I}$ is a *model* of a fuzzy KB *KB* iff $\mathcal{I}$ satisfies each fuzzy axiom in *KB*
- If *KB* has a model we say that *KB* is *satisfiable* (or *consistent*)
- Best entailment degree:

$$bed(KB, a{:}A) = m = \sup\{n \mid KB \models \langle a{:}A, n \rangle\} \ .$$

- viz.. in all models of *KB*, the degree of *a* being an instance of *A* is $\geq m$ and *m* is maximal

Meaning of P/N rules

- Each P-rule will tell us why an individual should be positive
- Each N-rule will tell us why an individual should be *non-positive*, i.e. should be negative
- Then, we use an aggregation operator to establish whether an individual is an instance of T or not (viz. is positive or negative) by combining the degree of being positive or negative via the @ operator

$$@(p, n) = \begin{cases} 1 & \text{if } p > n \\ 0 & \text{otherwise} \end{cases} \qquad (\star)$$

- For $a \in \mathsf{I}_{KB}$, $h_T(a)$ is called the classification prediction value of a

- Some more notions used here and there...
  - Hypothesis h_T covers an individual $a \in \mathsf{I}_{KB}$ iff $h_T(a) = 1$
  - $Cov(h_T)$ the set of covered individuals
  - h_P (resp. h_N) of P-rules (resp. N-rules) θ-covers, for $\theta \in (0, 1]$ an individual $a \in \mathsf{I}_{KB}$ iff $p_a \geq \theta$ (resp. $n_a \geq \theta$), and indicate with $Cov_\theta(h_P)$ (resp. $Cov_\theta(h_N)$) the set of covered individuals by h_P (resp. h_N)
  - Confidence degree or precision of $C \sqsubseteq X$ w.r.t. KB a set of positive examples P

$$cf(C \sqsubseteq X, KB, P) = \frac{|C|^{P}_{KB}}{|C|^{\mathsf{I}_{KB}}_{KB}}$$

  - Support of $C \sqsubseteq X$ w.r.t. KB

$$supp(C \sqsubseteq X, KB, \mathsf{I}) = \frac{|C|^{\mathsf{I}}_{KB}}{|\mathsf{I}|}$$

PN-OWL: A Two Stage Algorithm to Learn Fuzzy Concept Inclusions

P-Stage: Learn rules covering as many *Positives* as possible, without covering too many *False Positives* (increase *Recall*)

N-Stage: Learn rules covering as many *False Positives*, covering as few *Positives* as possible (increase *Precision*)

Decision: Take a decision depending on *Positive degree* versus *Negative degree*

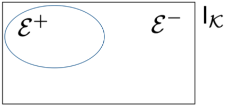
- Conceptually simple method
- Parametric wrt underlying rule learning method
- Surprisingly effective
- Examples of learnt P/N rules wrt Mammography ontology:

P-rule: (hasMargin some Ill-defined) and (hasShape some Irregular) and (hasAge some HighAge) SubClassOf P, 0.853

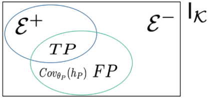
N-rule: (hasDensity some LowDensity) and (hasAge some MediumAge) and (hasBiRads some MediumBiRads) SubClassOf N, 1.0

Patient p102: $p_{p102} = 0.262$, $n_{p102} = 0.692$ and, thus, $@(0.262, 692) = 0$, i.e., “p102 does not have breast cancer”

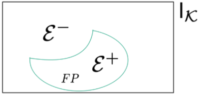
PN-OWL: Algorithm (Conceptually)



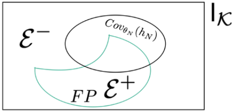
Init



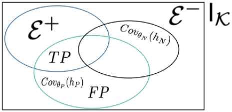
P-Stage: learn rules to cover positives



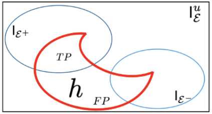
Prepare N-Stage: False Positives become new positives



N-Stage: learn rules to cover False Positives



After PN-Stage: coverings



Learned hypothesis after PN-Stage

Algorithm 1 PN-OWL

Input: A satisfiable crisp KB $\mathcal{K}$, a set $\mathcal{E}$ of training examples, set of parameters
Output: Stage hypotheses $h_P \cup h_N$ as by Eqs. 1-2.

// P-stage
$Pos \leftarrow \mathcal{E}^+$;
$Neg \leftarrow \mathcal{E}^-$;
$h_P \leftarrow$ FUZZYSTAGELEARNER(KB, P, Pos, Neg, θ_P, η_P); //P-Stage hypothesis h_P, i.e. set of axioms of the form $\langle C_i \sqsubseteq P, \alpha_i \rangle$, where P is a new concept name
if $h_P = \emptyset$ **then return** $\emptyset$; //Nothing learnt, exit
$Cov \leftarrow Cov_{\theta_P}(h_P)$; //P-stage Coverage
$TP \leftarrow Cov_{\theta_P}(h_P) \cap \mathcal{E}^+$; //True positives
$FP \leftarrow Cov_{\theta_P}(h_P) \cap \mathcal{E}^-$; //False positives
// Start building classifier h
$h \leftarrow h_P$; //As per Eq. 1
if $FP = \emptyset$ **then** //No N-stage, exit
    return h;
// N-stage
$Pos \leftarrow FP$;
$Neg \leftarrow \mathrm{I}_{KB} \setminus Pos$;
$h_N \leftarrow$ FUZZYSTAGELEARNER(KB, N, Pos, Neg, θ_N, η_N); //N-Stage hypothesis h_N, i.e. set of axioms of the form $\langle D_j \sqsubseteq N, \beta_j \rangle$, where N is a new concept name
// Build final classifier ensemble h
if $h_N = \emptyset$ **then** //No learning in N-stage, exit
    return h;
$h \leftarrow h \cup h_N$; //As per Eq. 2
return h;

Evaluation

- Datasets:

ontology	DL	class.	obj. prop.	data prop	ind.	target T	pos
NTN	$\mathcal{SHOIN}(\mathcal{D})$	51	29	9	723	ToLearn_Woman	46
Lymphography	$\mathcal{ALC}$	50	0	0	148	ToLearn	81
Mammographic	$\mathcal{ALC}(\mathcal{D})$	20	3	2	975	ToLearn	445
Malware	$\mathcal{ALH}(\mathcal{D})$	192	6	10	5669	malware	500
Iris	$\mathcal{ALEHF}(\mathcal{D})$	4	0	5	150	Iris-versicolor Iris-virginica	50 50
Wine	$\mathcal{ALEHF}(\mathcal{D})$	3	0	13	178	1 2 3	59 71 48
Wine Quality	$\mathcal{ALEHF}(\mathcal{D})$	7	0	11	6497	GoodRedWine	217
Yeast	$\mathcal{ALEHF}(\mathcal{D})$	11	0	8	1462	CYT	444

- Stratified k-Fold Cross Validation: to assess effectiveness

  Partitioning the dataset Examples are partitioned in k disjoint subsets $flds = \{fld_1, \ldots, fld_k\}$, called folds. In a stratified design, each fold fld_i contains roughly the same ratio of positive to negative examples as the full set of examples

  Training and testing The model is trained and tested k times using different training and test sets: at iteration i, the fold fld_i is used as test set and $flds \setminus fld_i$ as training set.

  Effectiveness evaluation After all iterations are complete, the effectiveness metrics are averaged over the k different test sets

Evaluation Results

Table: Results table. Metrics are averaged over the test sets of the five folds. For the multi-target datasets Iris and Wine, the values are macro-averaged over the targets. The ↑/↓ arrows mean that a higher/lower value of the metric is better.

Dataset	Algorithm	Prec ↑	Rec ↑	F1 ↑	MR ↓	%ΔF1	%ΔMR
NTN	Fuzzy FOIL-DL	0.661	0.513	0.548	0.053	**+80.47%**	**-98.11%**
	PN-OWL	**1.000**	**0.980**	**0.989**	**0.001**		
Lymphography	Fuzzy FOIL-DL	0.810	0.803	0.805	0.210	**+3.48%**	**-12.38%**
	PN-OWL	**0.836**	**0.841**	**0.833**	**0.184**		
Mammographic	Fuzzy FOIL-DL	0.737	0.692	0.710	0.256	**+10.56%**	**-19.14%**
	PN-OWL	**0.746**	**0.831**	**0.785**	**0.207**		
Malware	Fuzzy FOIL-DL	0.623	**0.830**	0.704	0.060	**+4.97%**	**-18.12%**
	PN-OWL	**0.701**	0.818	**0.739**	**0.049**		
Iris	Fuzzy FOIL-DL	0.886	0.910	0.890	0.077	**+4.16%**	**-41.18%**
	PN-OWL	**0.949**	0.910	**0.927**	**0.045**		
Wine	Fuzzy FOIL-DL	0.884	**0.971**	0.895	0.091	**+2.12%**	**-37.50%**
	PN-OWL	**0.933**	0.904	**0.914**	**0.057**		
Wine Quality	Fuzzy FOIL-DL	0.227	**0.917**	0.363	0.109	**+27.93%**	**-53.21%**
	PN-OWL	**0.365**	0.659	**0.464**	**0.051**		
YEAST	Fuzzy FOIL-DL	0.427	0.746	0.540	**0.382**	**+4.37%**	**+0.26%**
	PN-OWL	**0.432**	**0.815**	**0.564**	0.383		

Conclusion

- The overall lesson learnt with PN-OWL: provided one may find the appropriate balance between precision and recall, the N-stage may indeed provide a non-negligible contribution to improve the effectiveness of the classification process
- Unfortunately, searching the parameter space of PN-OWL for an optimum is time-consuming and a brute-force approach may likely not be feasible (at least not with our computational resources at hand) thus, we used a random grid search

Appendix

PN-OWL: parameters

Table: Some salient parameters of the PN-OWL algorithm.

param	description
θ_P	confidence threshold for positive rules of P-stage
θ_N	confidence threshold for negative rules of N-stage
η_P	non-positive coverage percentage threshold for positive rules of P-stage
η_N	non-positive coverage percentage threshold for negative rules of N-stage
c_P	maximal number of conjuncts for positive rules of P-stage
c_N	maximal number of conjuncts for negative rules of N-stage
d_P	maximal role depth for positive rules of P-stage
d_N	maximal role depth for negative rules of N-stage

We tested several configurations performing a random grid search over the large parameter-space $\langle \theta_P, \theta_N, \eta_P, \eta_N, c_P, c_N \rangle$. The maximal role depth has been set to an ontology-dependent threshold (usually, $d_P = d_N$) determined (manually) a priori through an inspection of the ontology.

The stage learner Fuzzy FOIL-$\mathcal{DL}$

- Invocations to FUZZYSTAGELEARNER at steps 4 and 17 in Algorithm 1 are calls to Fuzzy FOIL-$\mathcal{DL}$
- Essentially, Fuzzy FOIL-$\mathcal{DL}$ carries on inducing GCIs until as many positive examples are covered or nothing new can be learnt
- When a rule is induced (step 3), the positive examples still to be covered are updated (steps 9 and 10)
- To induce an axiom (step 3), LEARN-ONE-AXIOM is invoked (see Algorithm 3)

Algorithm 2 Fuzzy FOIL-$\mathcal{DL}$

Input: A satisfiable crisp KB $\mathcal{K}$, target concept name T not occurring in $\mathcal{K}$, a set P (resp. N) of positive (resp. negative) examples, confidence threshold $\theta \in (0, 1]$, negative coverage percentage $\eta \in [0, 1]$

Output: A hypothesis, i.e. a set $h_T = \{\langle C_i \sqsubseteq T, \delta_i \rangle | 1 \leq i \leq k\}$ of fuzzy $\mathcal{EL}(D)$ GCIs

1: $h_T \leftarrow \emptyset$, $Pos \leftarrow P$, $\phi \leftarrow \top \sqsubseteq T$;
2: **while** ($Pos \neq \emptyset$) **and** ($\phi \neq$ **null**) **do** //Loop until no improvement
3:     $\phi \leftarrow$ LEARN-ONE-AXIOM($KB, T, Pos, P, N, \theta, \eta$); //Learn one fuzzy $\mathcal{EL}(D)$ GCI of the form $C \sqsubseteq T$
4:     **if** $\phi \in h_T$ **then** //axiom already learnt
5:         $\phi \leftarrow$ **null**;
6:     **if** $\phi \neq$ **null then**
7:         $\delta \leftarrow cf(\phi, KB, P)$; //Compute confidence of ϕ, as per Eq. 37
8:         $h_T \leftarrow h_T \cup \{\langle \phi, \delta \rangle\}$; //Update hypothesis
9:         $Pos_\phi \leftarrow Pos \cap Cov(\langle \phi, \delta \rangle)$;//Positives covered by $\langle \phi, \delta \rangle$)
10:         $Pos \leftarrow Pos \setminus Pos_\phi$; //Update positives still to be covered
11: **return** h_T;

The algorithm LEARN-ONE-AXIOM

- In general terms it operates as follows:
  1. start from concept $\top$
  2. apply a refinement operator to find more specific fuzzy $\mathcal{EL}(D)$ concept candidates
  3. score them to choose the best candidate
  4. re-apply the refinement operator until a good candidate is found
  5. iterate the whole procedure until a satisfactory coverage of the positive examples is achieved

Algorithm 3 LEARN-ONE-AXIOM

Input: Satisfiable crisp KB $\mathcal{K}$, target $T \not\in \mathcal{K}$, set *Pos* of positive examples to be covered, training sets P, N of positive and negative examples, confidence threshold $\theta \in (0, 1]$, negative coverage percentage $\eta \in [0, 1]$

Output: A fuzzy $\mathcal{EL}(D)$ GCI of the form $C \sqsubseteq T$

$C \leftarrow \top$; //Start from $\top$
$\phi \leftarrow C \sqsubseteq T$;
while $C \neq$ **null do** //Loop until no improvement
    $C_{best} \leftarrow C$; $maxgain \leftarrow 0$;
    $\mathcal{C} \leftarrow \rho(C)$; //Compute all refinements of C
    for all $C' \in \mathcal{C}$ **do** //Compute the score of the refinements and select the best one
        $\phi' \leftarrow C' \sqsubseteq T$;
        $gain \leftarrow gain(\phi', \phi, KB, Pos)$;
        if $(gain > maxgain)$ **then**
            $maxgain \leftarrow gain$;
            $C_{best} \leftarrow C'$;
    if $C_{best} = C$ **then** //No improvement
        if $(cf(C_{best} \sqsubseteq T, KB, P) \geq \theta)$ **and** $supp(C_{best} \sqsubseteq T, KB, N) \leq \eta)$ **then break**;
    $C \leftarrow C_{best}$;
    $\phi \leftarrow C \sqsubseteq T$;
return ϕ;

The Downward Refinement Operator

- It takes as input a concept C and generates a more specific concept description candidates D
- $\mathbf{A}_{KB}$ is the set of all atomic concepts in KB
- $\mathbf{R}_{KB}$ is the set of all object properties in KB
- $\mathbf{S}_{KB}$ is the set of all numeric datatype properties in KB
- $\mathbf{B}_{KB}$ is the set of all Boolean datatype properties in KB
- $\mathbf{D}$ a fuzzy concrete domain
- For concepts C and D, let $C \sqsubseteq_{KB} D$ be a macro for $KB \models C \sqsubseteq D$, then $\sqsubset_{KB}$ is the strict sub-relation of $\sqsubseteq_{KB}$, i.e. if $C \sqsubseteq_{KB} D$ then $D \sqsubseteq_{KB} C$ does not hold

$$\rho(C) = \begin{cases}
\mathbf{A}_{KB} \cup \{\exists r.\top \mid r \in \mathbf{R}_{KB}\} \cup \\
\quad \{\exists s.\mathbf{d} \mid s \in \mathbf{S}_{KB}, \mathbf{d} \in \mathbf{D}\} \cup \\
\quad \{\exists s. =_b, \mid s \in \mathbf{B}_{KB}, b \in \{\mathbf{t}, \mathbf{f}\}\} & \text{if} \quad C = \top \\
\{A' \mid A' \in \mathbf{A}_{KB}, A' \sqsubset_{KB} A\} \cup \\
\quad \{A \sqcap A'' \mid A'' \in \rho(\top)\} & \text{if} \quad C = A \\
\{\exists r.D' \mid D' \in \rho(D)\} \cup \\
\quad \{(\exists r.D) \sqcap D'' \mid D'' \in \rho(\top)\} & \text{if} \quad C = \exists r.D, r \in \mathbf{R}_{KB} \\
\{(\exists s.d) \sqcap D \mid D \in \rho(\top)\} & \text{if} \quad C = \exists s.d, s \in \mathbf{S}_{KB}, \\
& \quad\; d \in \mathcal{D} \\
\{(\exists s. =_b) \sqcap D \mid D \in \rho(\top)\} & \text{if} \quad C = \exists s. =_b, s \in \mathbf{B}_{KB}, \\
& \quad\; b \in \{\mathbf{t}, \mathbf{f}\} \\
\{C_1 \sqcap \ldots C_i' \ldots \sqcap C_n \mid C_i' \in \rho(C_i)\} & \text{if} \quad C = C_1 \sqcap \ldots \sqcap C_n
\end{cases}$$

The Rule Scoring Function

- Information Gain: given a fuzzy $\mathcal{EL}(D)$ GCI ϕ of the form $C \sqsubseteq T$ chosen at the previous step, a KB *KB*, a set of positive examples *Pos* still to be covered and a candidate fuzzy $\mathcal{EL}(D)$ GCI ϕ' of the form $C' \sqsubseteq T$, then

$$gain(\phi', \phi, KB, Pos) = p * (log_2(cf(\phi', KB, Pos)) - log_2(cf(\phi, KB, Pos)))$$

  where $p = |C' \sqcap C|_{KB}^{Pos}$ is the fuzzy cardinality of positive examples in *Pos* covered by ϕ that are still covered by ϕ'
- The fuzzy cardinality or sigma-count of C w.r.t. *KB* and set of individuals I, denoted $|C|_{KB}^{\mathsf{I}}$, is defined as

$$|C|_{KB}^{\mathsf{I}} = \sum_{a \in \mathsf{I}} bed(KB, a{:}C) \ .$$