$\mathcal{ALC}$ Concept Learning with Refinement Operators

Jens Lehmann

Artificial Intelligence Institute
Computer Science Department
Technische Universität Dresden

September 27, 2006

Outline

1. the learning problem
2. refinement operators and their properties
3. the refinement operators $\rho_\downarrow$ and $\rho_\downarrow^{cl}$
4. conclusions and future work

Learning Problem

- goal: learn a concept definition from positive examples + negative examples + background knowledge
- we have a target concept name `Target` and a knowledge base $\mathcal{K}$ as background knowledge
- examples are of the form $\texttt{Target}(a)$, where a is an object
- let E^+ be the set of positive examples and E^- the set of negative examples
- we want to find a definition *Def* of the form $\texttt{Target} \equiv C$ such that for $\mathcal{K}' = \mathcal{K} \cup \{Def\}$ we have $\mathcal{K}' \models E^+$ and $\mathcal{K}' \not\models E^-$

Simple Example

$\texttt{Male} \equiv \neg\texttt{Female}$	$\texttt{Male}(\texttt{MARC})$
	$\texttt{Male}(\texttt{STEPHEN})$
$\texttt{hasChild}(\texttt{STEPHEN},\texttt{MARC})$	$\texttt{Male}(\texttt{JASON})$
$\texttt{hasChild}(\texttt{MARC},\texttt{ANNA})$	$\texttt{Male}(\texttt{JOHN})$
$\texttt{hasChild}(\texttt{JOHN},\texttt{MARIA})$	$\texttt{Female}(\texttt{ANNA})$
$\texttt{hasChild}(\texttt{ANNA},\texttt{JASON})$	$\texttt{Female}(\texttt{MARIA})$
	$\texttt{Female}(\texttt{MICHELLE})$

positive: $\{\texttt{STEPHEN}, \texttt{MARC}, \texttt{JOHN}\}$

negative: $\{\texttt{JASON}, \texttt{ANNA}, \texttt{MARIA}, \texttt{MICHELLE}\}$

learned concept: $\texttt{Male} \sqcap \exists\texttt{hasChild}.\top$

Application Areas

Why is it useful to learn in DLs?

- may have similar applications like ILP (Inductive Logic Programming) approaches for learning horn clauses e.g. in biology and medicine
- incremental ontology learning in context of OWL and the Semantic Web – the lack of ontologies is a bottleneck in the Semantic Web

Solving the Learning Problem

Two learning approaches in my thesis:

1. usage of refinement operators combined with a search heuristic (main topic of presentation)
    - search in the space of concepts ordered by subsumption
    - similar to traditional Inductive Logic Programming methods
2. usage of Genetic Programming
    - has been used (with non-standard operators) in ILP to induce logic programs
    - in thesis, insights about refinement operators were used to improve standard GP, resulting in a hybrid approach
    - proposed extensions: learning from uncertain examples, concept invention

Refinement Operators - Definitions

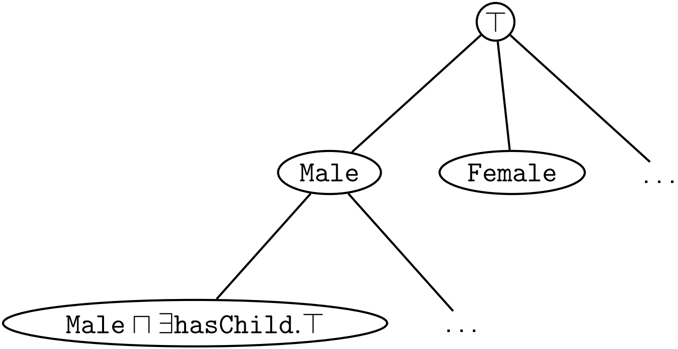
- consider quasi-ordered space $(S, \preceq)$, i.e. $\preceq$ is reflexive and transitive
- downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

$$C' \in \rho(C) \text{ implies } C' \preceq C \quad (C \preceq C')$$

- refinement operator in the quasi-ordered space $(\mathcal{ALC}, \sqsubseteq_{\mathcal{T}})$ is called an $\mathcal{ALC}$ refinement operator
- a refinement chain of an $\mathcal{ALC}$ refinement operator ρ from a concept C to a concept D is a finite sequence $C_0, C_1, \ldots, C_n$ of concepts, such that
  $C = C_0, C_1 \in \rho(C_0), C_2 \in \rho(C_1), \ldots, C_n \in \rho(C_{n-1}), D = C_n$
- instead of $D \in \rho(C)$ we often write $C \rightsquigarrow_\rho D$, e.g.
  $\top \rightsquigarrow_\rho \texttt{Male} \rightsquigarrow_\rho \texttt{Male} \sqcap \exists \texttt{hasChild}.\top$

Learning with Refinement Operators

- refinement operator can be used to span up a search tree
- refinement operator + search heuristic = learning algorithm



Properties of Refinement Operators

An $\mathcal{ALC}$ refinement operator ρ is called

- finite iff $\rho(C)$ is finite for any concept C.
- redundant iff there exist two different refinement chains from a concept C to a concept D.
- proper iff for any concepts C and D, $D \in \rho(C)$ implies $C \not\equiv_{\mathcal{T}} D$.

An $\mathcal{ALC}$ downward refinement operator is called

- complete iff for any concepts C and D with $C \sqsubset_{\mathcal{T}} D$ we can reach a concept E with $E \equiv_{\mathcal{T}} C$ from D by ρ.
- weakly complete iff for any concept C with $C \sqsubset_{\mathcal{T}} \top$ we can reach a concept E with $E \equiv_{\mathcal{T}} C$ from $\top$ by ρ.

Theorem about Properties of $\mathcal{ALC}$ Refinement Operators

Theorem (properties of $\mathcal{ALC}$ refinement operators)

Considering these properties, the following are maximal sets of properties of $\mathcal{ALC}$ refinement operators:

1. {*weakly complete*, *complete*, *finite*}
2. {*weakly complete*, *complete*, *proper*}
3. {*weakly complete*, *non-redundant*, *finite*}
4. {*weakly complete*, *non-redundant*, *proper*}
5. {*non-redundant*, *finite*, *proper*}

$\rho_\downarrow$ part 1 of 2

$$\rho_\downarrow(C) = \begin{cases} \{\bot\} \cup \rho'_\downarrow(C) & \text{if } C = \top \\ \rho'_\downarrow(C) & \text{otherwise} \end{cases}$$

$$\rho'_\downarrow(C) = \begin{cases} \{C_1 \sqcap \dots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \dots \sqcap C_n & \text{if } C = C_1 \sqcap \dots \sqcap C_n \\ \quad \mid D \in \rho'_\downarrow(C_i), 1 \leq i \leq n\} & \\ \{C_1 \sqcup \dots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \dots \sqcup C_n & \text{if } C = C_1 \sqcup \dots \sqcup C_n \\ \quad \mid D \in \rho'_\downarrow(C_i), 1 \leq i \leq n\} & \\ \cup \{C \sqcap D \mid D \in \rho'_\downarrow(\top)\} & \\ \{A' \mid A' \sqsubset_{\mathcal{T}} A, A' \in N_C, \text{ there is} & \text{if } C = A \, (A \in N_C) \\ \text{ no } A'' \in N_C \text{ with } A' \sqsubset_{\mathcal{T}} A'' \sqsubset_{\mathcal{T}} A\} & \\ \cup \{C \sqcap D \mid D \in \rho'_\downarrow(\top)\} & \\ \{\neg A' \mid A \sqsubset_{\mathcal{T}} A', A' \in N_C, \text{ there is} & \text{if } C = \neg A \, (A \in N_C) \\ \text{ no } A'' \in N_C \text{ with } A \sqsubset_{\mathcal{T}} A'' \sqsubset_{\mathcal{T}} A'\} & \\ \cup \{C \sqcap D \mid D \in \rho'_\downarrow(\top)\} & \end{cases}$$

$\rho_\downarrow$ part 2 of 2

$$\rho'_\downarrow(C) = \begin{cases} \{\exists r.E \mid E \in \rho'_\downarrow(D)\} & \text{if } C = \exists r.D \\ \cup \{C \sqcap D \mid D \in \rho'_\downarrow(\top)\} & \\ \{\forall r.E \mid E \in \rho'_\downarrow(D)\} \cup \{C \sqcap D \mid D \in \rho'_\downarrow(\top)\} & \text{if } C = \forall r.D \\ \cup \{\forall r.\bot \mid D = A \in N_C & \\ \text{ and there is no } A' \in N_C \text{ with } \bot \sqsubset A' \sqsubset A\} & \\ \emptyset & \text{if } C = \bot \\ \{D \mid D \in M\} & \text{if } C = \top \\ \cup \{D \sqcup E \mid D \in M, E \in \rho'_\downarrow(\top)\} & \\ M = & \\ \{A \mid A \in N_C, & \\ \quad \text{there is no } A' \in N_C \text{ with } A \sqsubset_{\mathcal{T}} A' \sqsubset_{\mathcal{T}} \top\} & \\ \cup \{\neg A \mid A \in N_C, & \\ \quad \text{there is no } A' \in N_C \text{ with } \bot \sqsubset_{\mathcal{T}} A' \sqsubset_{\mathcal{T}} A\} & \\ \cup \{\exists r.\top \mid r \in N_R\} & \\ \cup \{\forall r.C \mid r \in N_R, C \in \rho'_\downarrow(\top)\} & \end{cases}$$

Completeness of $\rho_\downarrow$

Proposition (completeness of $\rho_\downarrow$)

$\rho_\downarrow$ is complete.

Proof Idea:

- first show weak completeness:
  - a set $S_\downarrow$ of $\mathcal{ALC}$ concepts was defined (see thesis for the definition of $S_\downarrow$)
  - for every $\mathcal{ALC}$ concept there exists an equivalent concept in $S_\downarrow$
  - all concepts in $S_\downarrow$ can be reached by $\rho_\downarrow$ from $\top$
- prove completeness using the weak completeness result

Infiniteness of $\rho_\downarrow$

- $\rho_\downarrow$ is infinite, e.g. there are infinitely many refinement steps of the form:

$$\top \leadsto_{\rho_\downarrow} \underbrace{\forall \texttt{hasChild}.\ldots.\forall \texttt{hasChild}}_{\text{arbitrarily often}}.\texttt{Male}$$

- solution: we only consider refinements up to length n of concepts (there are only finitely many of these)
- n is initially set to 0 and increased by the learning algorithm as needed

Properness

- $\rho_\downarrow$ is not proper: $\top \rightsquigarrow_{\rho_\downarrow} \exists\mathtt{hasChild}.\top \sqcup \forall\mathtt{hasChild}.\mathtt{Male}$
- idea: consider the closure $\rho_\downarrow^{cl}$ of $\rho_\downarrow$:
  $D \in \rho_\downarrow^{cl}(C)$ iff there exists a refinement chain

$$C \rightsquigarrow_{\rho_\downarrow} C_1 \rightsquigarrow_{\rho_\downarrow} \ldots \rightsquigarrow_{\rho_\downarrow} C_n = D$$

  such that $C \not\equiv D$ and $C_i \equiv C$ for $i \in \{1, \ldots, n-1\}$

Proposition

For any concept C in negation normal form and any natural number n the set

$$\{D \mid D \in \rho_\downarrow^{cl}(C), |D| \leq n\}$$

can be computed in finite time.

Redundancy

$\rho_{\downarrow}^{cl}$ is redundant:

$$\begin{tikzcd}
& \forall r_1.A_1 \sqcup \forall r_2.A_1 \arrow[r, rightsquigarrow, "\rho_{\downarrow}"'] \arrow[d, rightsquigarrow, "\rho_{\downarrow}"'] & \forall r_1.(A_1 \sqcap A_2) \sqcup \forall r_2.A_1 \arrow[d, rightsquigarrow, "\rho_{\downarrow}"'] \\
& \forall r_1.A_1 \sqcup \forall r_2.(A_1 \sqcap A_2) \arrow[r, rightsquigarrow, "\rho_{\downarrow}"'] & \forall r_1.(A_1 \sqcap A_2) \sqcup \forall r_2.(A_1 \sqcap A_2)
\end{tikzcd}$$

- redundancies should be detected by the learning algorithm
- result in thesis: we can check whether an occurring concept is redundant with respect to a search tree in polynomial time

Conclusions

- first full analysis of theoretical properties of refinement operators for Description Logics
- complete, proper operator was defined and ways to handle infinity and redundancy were shown
- theoretical results ensure that the learning algorithm is close to the best we can hope for
- result: if a solution exists, then the algorithm terminates in finite time and finds a solution

Related Work (Excerpt)

Badea, L. and Nienhuys-Cheng, S.-H. (2000). A refinement operator for description logics. *Proceedings of the 10th International Conference on Inductive Logic Programming*

Esposito, F., Fanizzi, N., Iannone, L., Palmisano, I., and Semeraro, G. (2004). Knowledge-intensive induction of terminologies from metadata. *Proceedings of the Third International Semantic Web Conference*

Iannone, L. and Palmisano, I. (2005). An algorithm based on counterfactuals for concept learning in the semantic web. *Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*

Future Work

- implement learning algorithms (refinement operator based approach with a search heuristic, Genetic Programming approach)
- create benchmarks
- embed learning algorithm in ontology editor
- extend theory to other description languages and OWL

Thank you for your attention.