Hybrid Learning of Ontology Classes

Jens Lehmann

UNIVERSITÄT LEIPZIG

July 18, 2007

Outline

1. Introduction to Description Logics, OWL, and the Learning Problem
2. Solving the Learning Problem with Genetic Programming (GP)
3. Genetic Refinement Operators
4. Preliminary Evaluation
5. Conclusions & Future Work

Outline

1. Introduction to Description Logics, OWL, and the Learning Problem
2. Solving the Learning Problem with Genetic Programming (GP)
3. Genetic Refinement Operators
4. Preliminary Evaluation
5. Conclusions & Future Work

Introduction to Description Logics

- Description Logics (DL) is the name of a family of languages for knowledge representation
- fragment of first order predicate logic
- less expressive power than predicate logic, but decidable inference problems
- intuitive variable free syntax
- basis of the ontology language OWL (W3C recommendation)
- OWL ontology convertable to DL knowledge base and vice versa

$$
\begin{aligned}
\texttt{Woman} &\equiv \texttt{Person} \sqcap \texttt{Female} \\
\texttt{Man} &\equiv \texttt{Person} \sqcap \neg\texttt{Female} \\
\texttt{Mother} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasChild}.\top \\
\texttt{Person} &\equiv \texttt{Man} \sqcup \texttt{Woman} \\
\bot &\equiv \texttt{Male} \sqcap \texttt{Female}
\end{aligned}
$$

$\mathcal{ALC}$ Description Logic knowledge base

TBox - terminological knowledge

The Learning Problem in DLs

$$\begin{aligned}
\texttt{Woman} &\equiv \texttt{Person} \sqcap \texttt{Female} \\
\texttt{Man} &\equiv \texttt{Person} \sqcap \neg\texttt{Female} \\
\texttt{Mother} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasChild}.\top \\
\texttt{Person} &\equiv \texttt{Man} \sqcup \texttt{Woman} \\
\bot &\equiv \texttt{Male} \sqcap \texttt{Female}
\end{aligned}$$

$\texttt{hasChild(STEPHEN,MARC)}$
$\texttt{hasChild(MARC,ANNA)}$
$\texttt{hasChild(JOHN,MARIA)}$
$\texttt{hasChild(ANNA,JASON)}$

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

$\mathcal{ALC}$ Description Logic knowledge base
ABox - assertional knowledge

The Learning Problem in DLs

$$
\begin{aligned}
\mathtt{Woman} &\equiv \mathtt{Person} \sqcap \mathtt{Female} \\
\mathtt{Man} &\equiv \mathtt{Person} \sqcap \neg\mathtt{Female} \\
\mathtt{Mother} &\equiv \mathtt{Woman} \sqcap \exists \mathtt{hasChild}.\top \\
\mathtt{Person} &\equiv \mathtt{Man} \sqcup \mathtt{Woman} \\
\bot &\equiv \mathtt{Male} \sqcap \mathtt{Female}
\end{aligned}
$$

$\mathtt{hasChild}(\mathtt{STEPHEN},\mathtt{MARC})$
$\mathtt{hasChild}(\mathtt{MARC},\mathtt{ANNA})$
$\mathtt{hasChild}(\mathtt{JOHN},\mathtt{MARIA})$
$\mathtt{hasChild}(\mathtt{ANNA},\mathtt{JASON})$

$\mathtt{Male}(\mathtt{JOHN})$
$\mathtt{Male}(\mathtt{MARC})$
$\mathtt{Male}(\mathtt{STEPHEN})$
$\mathtt{Male}(\mathtt{JASON})$
$\mathtt{Female}(\mathtt{MICHELLE})$
$\mathtt{Female}(\mathtt{ANNA})$
$\mathtt{Female}(\mathtt{MARIA})$

positive examples: $\{\mathtt{STEPHEN}, \mathtt{MARC}, \mathtt{JOHN}\}$

negative examples: $\{\mathtt{JASON}, \mathtt{ANNA}, \mathtt{MARIA}, \mathtt{MICHELLE}\}$

The Learning Problem in DLs

$$
\begin{aligned}
\texttt{Woman} &\equiv \texttt{Person} \sqcap \texttt{Female} \\
\texttt{Man} &\equiv \texttt{Person} \sqcap \neg\texttt{Female} \\
\texttt{Mother} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasChild}.\top \\
\texttt{Person} &\equiv \texttt{Man} \sqcup \texttt{Woman} \\
\bot &\equiv \texttt{Male} \sqcap \texttt{Female}
\end{aligned}
$$

$\texttt{hasChild}(\texttt{STEPHEN},\texttt{MARC})$
$\texttt{hasChild}(\texttt{MARC},\texttt{ANNA})$
$\texttt{hasChild}(\texttt{JOHN},\texttt{MARIA})$
$\texttt{hasChild}(\texttt{ANNA},\texttt{JASON})$

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

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

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

possible solution: $\texttt{Target} \equiv \texttt{Male} \sqcap \exists\texttt{hasChild}.\top$

Outline

1. Introduction to Description Logics, OWL, and the Learning Problem
2. Solving the Learning Problem with Genetic Programming (GP)
3. Genetic Refinement Operators
4. Preliminary Evaluation
5. Conclusions & Future Work

Genetic Programming (GP)

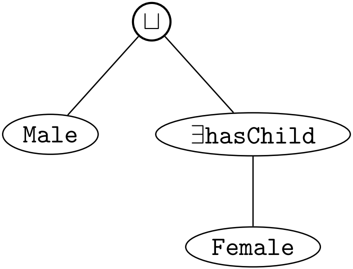
Algorithm (Genetic Programming)

- *create population*
- *while the termination criterion is not met:*
  - *select a subset of the population based on their fitness*
  - *produce offspring using genetic operators on selected individuals*
  - *create a new population from the old one and the offspring*

- genetic operators: crossover, mutation, editing
- selection: rank selection, FPS, tournament selection
- tree representation common in GP

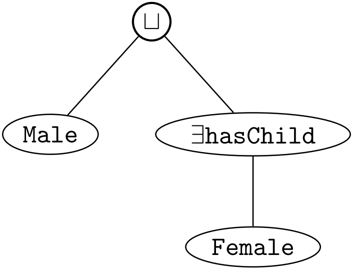
Applying Standard GP

- representing $\mathcal{ALC}$ concepts:
  - terminal set
    $T = N_C \cup \{\top, \bot\}$
  - function set
    $F = \{\sqcup, \sqcap, \neg\} \cup \{\forall r \mid r \in N_R\}$
    $\cup \{\exists r \mid r \in N_R\}$



Applying Standard GP

- representing $\mathcal{ALC}$ concepts:
  - terminal set
    $T = N_C \cup \{\top, \bot\}$
  - function set
    $F = \{\sqcup, \sqcap, \neg\} \cup \{\forall r \mid r \in N_R\}$
    $\cup \{\exists r \mid r \in N_R\}$



- possible fitness function:

$$f_{\mathcal{K}}(C) = -\frac{|E^+ \setminus pos_{\mathcal{K}}(C)| + |neg_{\mathcal{K}}(C)|}{|E^+| + |E^-|} - a \cdot |C| \quad (0 < a < 1)$$

  - $pos_{\mathcal{K}}(C)$ ... set of covered positive examples
  - $neg_{\mathcal{K}}(C)$ ... set of covered negative examples
  - a ... concept length penalty

Advantages&Disadvantages of the Standard GP Approach

- Tree encoding and fitness measurement are sufficient to apply GP!

Advantages&Disadvantages of the Standard GP Approach

- Tree encoding and fitness measurement are sufficient to apply GP!
- Advantages:
  - very flexible learning method (can handle other description languages)
  - parallelisable algorithms
  - GP is robust to noise

Advantages&Disadvantages of the Standard GP Approach

- Tree encoding and fitness measurement are sufficient to apply GP!
- Advantages:
  - very flexible learning method (can handle other description languages)
  - parallelisable algorithms
  - GP is robust to noise
- Disadvantages:
  - crossover operator too destructive: small syntactic changes - drastic semantic changes
  - does not use all of the available background knowledge: no exploitation of the subsumption hierarchy of concepts

Outline

1. Introduction to Description Logics, OWL, and the Learning Problem
2. Solving the Learning Problem with Genetic Programming (GP)
3. Genetic Refinement Operators
4. Preliminary Evaluation
5. Conclusions & Future Work

Refinement Operators

- idea: combine refinement operators and GP
- definition of refinement operators:
  - consider quasi-ordered space $(\mathcal{ALC}, \sqsubseteq)$
  - $\mathcal{ALC}$ downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

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

  - example: $\top$

Refinement Operators

- idea: combine refinement operators and GP
- definition of refinement operators:
  - consider quasi-ordered space $(\mathcal{ALC}, \sqsubseteq)$
  - $\mathcal{ALC}$ downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

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

  - example: $\top \rightsquigarrow \texttt{Person}$

Refinement Operators

- idea: combine refinement operators and GP
- definition of refinement operators:
  - consider quasi-ordered space $(\mathcal{ALC}, \sqsubseteq)$
  - $\mathcal{ALC}$ downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

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

  - example: $\top \rightsquigarrow \texttt{Person} \rightsquigarrow \texttt{Person} \sqcap \exists \texttt{takesPartIn.Conference}$

Refinement Operators

- idea: combine refinement operators and GP
- definition of refinement operators:
  - consider quasi-ordered space $(\mathcal{ALC}, \sqsubseteq)$
  - $\mathcal{ALC}$ downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

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

  - example: $\top \rightsquigarrow \texttt{Person} \rightsquigarrow \texttt{Person} \sqcap \exists \texttt{takesPartIn.Conference}$
- refinement operators ...

Refinement Operators

- idea: combine refinement operators and GP
- definition of refinement operators:
  - consider quasi-ordered space $(\mathcal{ALC}, \sqsubseteq)$
  - $\mathcal{ALC}$ downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$:

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

  - example: $\top \rightsquigarrow \texttt{Person} \rightsquigarrow \texttt{Person} \sqcap \exists\texttt{takesPartIn.Conference}$
- refinement operators ...
  - ... can make use of the generality order of concepts w.r.t. $\mathcal{K}$
  - ... are less destructive w.r.t. the semantics of a concept
  - ... can use the (precomputed) subsumption hierarchy

Genetic Refinement Operators

- What distinguishes refinement and genetic operators?
  - refinement operators map one concept to many concepts
  - refinement operators are either downward or upward operators

Genetic Refinement Operators

- What distinguishes refinement and genetic operators?
  - refinement operators map one concept to many concepts
  - refinement operators are either downward or upward operators
- solution: Genetic Refinement Operators

$$\phi_{\mathcal{K}}(C) = \begin{cases} \text{rand}(\phi_{\downarrow}(C)) & \text{with probability } \dfrac{\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}}{1+\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}} \\ \text{rand}(\phi_{\uparrow}(C)) & \text{with probability } \dfrac{1-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}}{1+\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}} \end{cases}$$

Genetic Refinement Operators

- What distinguishes refinement and genetic operators?
  - refinement operators map one concept to many concepts
  - refinement operators are either downward or upward operators
- solution: Genetic Refinement Operators

$$\phi_{\mathcal{K}}(C) = \begin{cases} \text{rand}(\phi_{\downarrow}(C)) & \text{with probability } \dfrac{\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}}{1+\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}} \\ \text{rand}(\phi_{\uparrow}(C)) & \text{with probability } \dfrac{1-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}}{1+\frac{|neg_{\mathcal{K}}(C)|}{|E^-|}-\frac{|pos_{\mathcal{K}}(C)|}{|E^+|}} \end{cases}$$

- we created a complete and proper operator based on a full property analysis [Lehmann et. al, ILP 2007]

Outline

1. Introduction to Description Logics, OWL, and the Learning Problem
2. Solving the Learning Problem with Genetic Programming (GP)
3. Genetic Refinement Operators
4. Preliminary Evaluation
5. Conclusions & Future Work

Evaluation - Uncle Problem

- learn definition of uncle from FORTE family data set (337 assertions, 86 examples)
- problem is challenging - relatively complex solution and no search space restrictions

possible solution:

$$\mathtt{Uncle} \equiv \mathtt{Male} \sqcap (\exists\, \mathtt{sibling}.\exists\, \mathtt{parent}.\top \sqcup \exists\, \mathtt{married}.\exists\, \mathtt{sibling}.\exists\, \mathtt{parent}.\top)$$

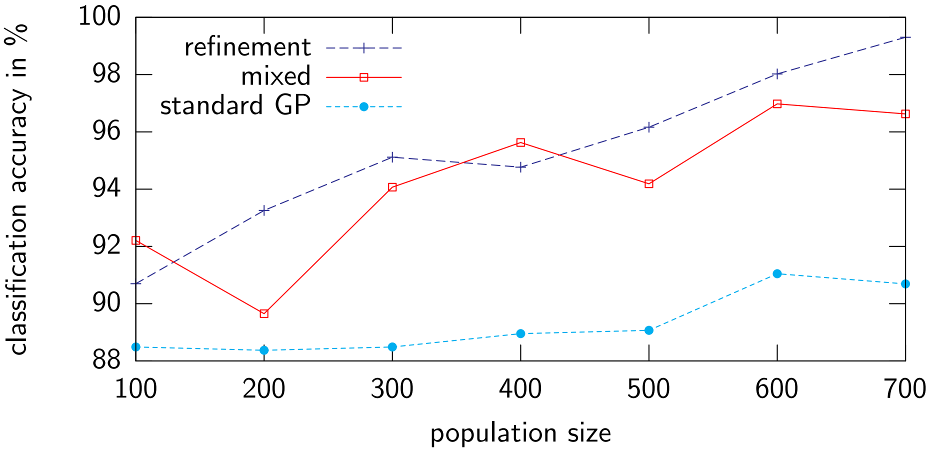
Evaluation - Uncle Problem

- learn definition of uncle from FORTE family data set (337 assertions, 86 examples)
- problem is challenging - relatively complex solution and no search space restrictions
- compare against state of the art DL learning system YinYang
- compare improvement over standard GP

possible solution:

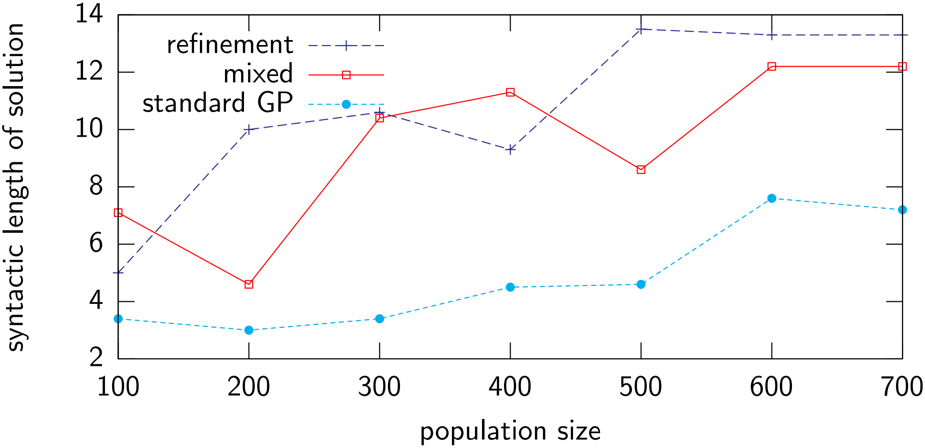
$$\texttt{Uncle} \equiv \texttt{Male} \sqcap (\exists\, \texttt{sibling}.\exists\, \texttt{parent}.\top \sqcup \exists\, \texttt{married}.\exists\, \texttt{sibling}.\exists\, \texttt{parent}.\top)$$

Evaluation - Accuracy



YinYang: 73.5%

Evaluation - Concept Length



YinYang: 12.2

Outline

Contributions to the State of the Art

- first time to apply evolutionary techniques to learning problem in DLs
- first framework for combining refinement operators and GP directly
- creation of a concrete operator based on a full property analysis
- implemented in a system called DL-Learner and shown to be feasible in a preliminary evaluation

Future Work

- more evaluation examples, e.g. asses performance on noisy or inconsistent data
- create (more) benchmarks to assess scalability and enable easier comparison between different algorithms
- tests on real world data, e.g. DBpedia
- embed learning algorithm in ontology editor e.g. OntoWiki
- extend algorithm to other description languages (cardinality restrictions, datatype integer)

Thank you for your attention.

contact:
`lehmann@informatik.uni-leipzig.de`