Learning Sentences and Assessments in Probabilistic Description Logics

José Ochoa Luna[1] Kate Revoredo[2] Fabio Cozman[1]

Decision Making Lab.
Escola Politécnica - Universidade de São Paulo

Departamento de Informática Aplicada
Unirio

URSW 2010

Outline

Motivation

- Representation of uncertainty in the semantic Web can be favoured by the use of learning techniques

Motivation

- Representation of uncertainty in the semantic Web can be favoured by the use of learning techniques
- Caveats in syntax and semantics in PDL have prevented them from spreading into several domains

Motivation

- Representation of uncertainty in the semantic Web can be favoured by the use of learning techniques
- Caveats in syntax and semantics in PDL have prevented them from spreading into several domains
- It can be hard to elicit the probability component of a particular set of sentences

Motivation

- Representation of uncertainty in the semantic Web can be favoured by the use of learning techniques
- Caveats in syntax and semantics in PDL have prevented them from spreading into several domains
- It can be hard to elicit the probability component of a particular set of sentences
- Focus in CR$\mathcal{ALC}$ language

Previous Efforts

Focus on Concept definitions

Using Noisy-OR classifiers

Previous Efforts

Focus on Concept definitions

Using Noisy-OR classifiers

Focus on Probabilistic inclusions

$C \equiv A \cup B \rightarrow P(C|A \cup B), P(C|A), P(C|B)$

Idea

Combined approach $\rightarrow$ an algorithm for learning concept definitions and probabilistic inclusions at once

Idea

Combined approach → an algorithm for learning concept definitions and probabilistic inclusions at once

Mostly based on inductive logic programming techniques with a probabilistic twist

Idea

Combined approach $\rightarrow$ an algorithm for learning concept definitions and probabilistic inclusions at once

Mostly based on inductive logic programming techniques with a probabilistic twist

- A search for the best concept description is performed.

Idea

Combined approach → an algorithm for learning concept definitions and probabilistic inclusions at once

Mostly based on inductive logic programming techniques with a probabilistic twist

- A search for the best concept description is performed.
- A decision is made as to whether to consider the concept definition found or to insert a probabilistic inclusion based on this concept

Probabilistic Description Logic CR$\mathcal{ALC}$

- CR$\mathcal{ALC}$ is a probabilistic extension of the DL $\mathcal{ALC}$.

Probabilistic Description Logic CR$\mathcal{ALC}$

- CR$\mathcal{ALC}$ is a probabilistic extension of the DL $\mathcal{ALC}$.
- The following constructors are available in $\mathcal{ALC}$: *conjunction* ($C \sqcap D$), *disjunction* $C \sqcup D$, *negation* ($\neg C$), *existential* restriction ($\exists r.C$), and value restriction ($\forall r.C$).

Probabilistic Inclusions and their Semantics

- $P(\mathrm{A}|\mathrm{B}) = \alpha$

Probabilistic Inclusions and their Semantics

- $P(\mathrm{A}|\mathrm{B}) = \alpha$
- $\forall x \in \mathcal{D} \;:\; P(\mathrm{A(x)}|\mathrm{B(x)}) = \alpha$

Probabilistic Inclusions and their Semantics

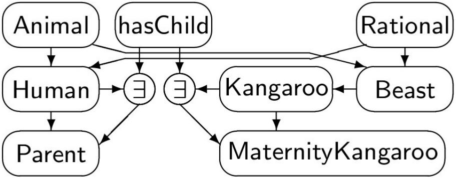
- $P(\mathrm{A}|\mathrm{B}) = \alpha$
- $\forall x \in \mathcal{D} \; : \; P(\mathrm{A}(\mathrm{x})|\mathrm{B}(\mathrm{x})) = \alpha$
- $P(\mathrm{Professor}(\mathrm{Maria})|\mathrm{Researcher}(\mathrm{Maria})) = 0.4$

Example

$P(\text{Animal}) = 0.9$,
$P(\text{Rational}) = 0.6$,
$P(\text{hasChild}) = 0.3$,
$\text{Human} \equiv \text{Animal} \sqcap \text{Rational}$,
$\text{Beast} \equiv \text{Animal} \sqcap \neg\text{Rational}$,
$\text{Parent} \equiv$
$\text{Human} \sqcap \exists\text{hasChild.Human}$,
$P(\text{Kangaroo}|\text{Beast}) = 0.4$,
$P(\text{Kangaroo}|\neg\text{Beast}) = 0.0$,
$\text{MaternityKangaroo} \equiv$
$\text{Kangaroo} \sqcap \exists\text{hasChild.Kangaroo}$

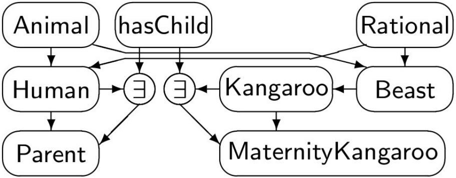
Example

$P(\text{Animal}) = 0.9$,
$P(\text{Rational}) = 0.6$,
$P(\text{hasChild}) = 0.3$,
$\text{Human} \equiv \text{Animal} \sqcap \text{Rational}$,
$\text{Beast} \equiv \text{Animal} \sqcap \neg\text{Rational}$,
$\text{Parent} \equiv$
$\text{Human} \sqcap \exists\text{hasChild.Human}$,
$P(\text{Kangaroo}|\text{Beast}) = 0.4$,
$P(\text{Kangaroo}|\neg\text{Beast}) = 0.0$,
$\text{MaternityKangaroo} \equiv$
$\text{Kangaroo} \sqcap \exists\text{hasChild.Kangaroo}$



Example

$P(\text{Animal}) = 0.9$,
$P(\text{Rational}) = 0.6$,
$P(\text{hasChild}) = 0.3$,
$\text{Human} \equiv \text{Animal} \sqcap \text{Rational}$,
$\text{Beast} \equiv \text{Animal} \sqcap \neg\text{Rational}$,
$\text{Parent} \equiv$
$\text{Human} \sqcap \exists\text{hasChild.Human}$,
$P(\text{Kangaroo}|\text{Beast}) = 0.4$,
$P(\text{Kangaroo}|\neg\text{Beast}) = 0.0$,
$\text{MaternityKangaroo} \equiv$
$\text{Kangaroo} \sqcap \exists\text{hasChild.Kangaroo}$



Inference

$$P(\text{Parent}(0)|\text{Human}(0)) = 0.232$$

Learning Description Logics

Goal

Find a correct concept with respect to given examples. A sound concept definition for Target must cover all positive examples and none of the negative examples

Learning Description Logics

Goal

Find a correct concept with respect to given examples. A sound concept definition for Target must cover all positive examples and none of the negative examples

- YINYANG

Learning Description Logics

Goal

Find a correct concept with respect to given examples. A sound concept definition for Target must cover all positive examples and none of the negative examples

- YINYANG
- DL-FOIL

Learning Description Logics

Goal

Find a correct concept with respect to given examples. A sound concept definition for Target must cover all positive examples and none of the negative examples

- YINYANG
- DL-FOIL
- DL-Learner

Learning Steps

Refinement Operators

Generalization and specialization $\rightarrow$ θ-subsumption

Learning Steps

Refinement Operators

Generalization and specialization $\rightarrow$ θ-subsumption

Score Function

$\mathcal{K} \cup C \models e$ (instance checking)

Learning Steps

Refinement Operators

Generalization and specialization $\rightarrow$ θ-subsumption

Score Function

$\mathcal{K} \cup C \models e$ (instance checking)

Search Algorithm

FOIL-based, genetic algorithms, horizontal expansion

Assumption

Deterministic

Father $\equiv$ Male $\sqcap$ hasChild.$\top$

Assumption

Deterministic

$\text{Father} \equiv \text{Male} \sqcap \text{hasChild}.\top$

Probabilistic

$P(\text{FlyingBird}|\text{Bird}) = \alpha$

Assumption

Deterministic

Father $\equiv$ Male $\sqcap$ hasChild.$\top$

Probabilistic

P(FlyingBird|Bird) $= \alpha$

Assumption

Negative and positive examples underlie the choice of either a concept definition or a probabilistic inclusion

Proposal

- We expect to find concepts covering all positive examples which is not always possible $\rightarrow$ uncertainty

Proposal

- We expect to find concepts covering all positive examples which is not always possible $\rightarrow$ uncertainty
- When we are unable to find a concept definition that covers all positive examples we assume such hypothesis as candidates to be a probabilistic inclusion

Proposal II

Refinement Operators

Previous refinement operators

Proposal II

Refinement Operators

Previous refinement operators

Probabilistic score function

$$cover(e, \mathcal{K}, C) = P(e|\mathcal{K}, C).$$

Proposal II

Refinement Operators

Previous refinement operators

Probabilistic score function

$$cover(e, \mathcal{K}, C) = P(e|\mathcal{K}, C).$$

Search

- We start by searching a deterministic concept

Proposal II

Refinement Operators

Previous refinement operators

Probabilistic score function

$$cover(e, \mathcal{K}, C) = P(e|\mathcal{K}, C).$$

Search

- We start by searching a deterministic concept
- If after a set of iterations the score of the best candidate is below a given threshold, a search for a probabilistic inclusion is started

The Algorithm

Require: an initial knowledge base $\mathcal{K} =< \mathcal{T}, \mathcal{A} >$ and a training set E.

SearchTree with a node $\{C = \top, h = 0\}$
repeat
    choose node $N = \{C, h\}$ with highest probabilistic score in SearchTree
    expand node to length $h + 1$:
    add all nodes $D \in$ (refinementOperator(C)) with lenght $=h + 1$
    learn parameters for all nodes D
    $N = \{C, h + 1\}$
    expand alternative nodes according to horizontal expansion factor and $h + 1$[18]
until stopping criterion
$N' =$ best node in SearchTree
if score(N') $>$ threshold **then**
    return deterministic concept $C' \in N'$
else
    call ProbabilisticInclusion(SearchTree)
end if

Algorithm 1: Algorithm for learning probabilistic terminologies.

Probabilistic Inclusion Algorithm

Require: SearchTree previously computed

for each pair of candidates C_i, C_j in first k nodes of the SearchTree **do**
    compute the conditional mutual information $I(C_i, C_j|T)$
end for
build an undirected graph in which vertices are the k candidates
annotate the weight of an edge connecting C_i to C_j by the $I(C_i, C_j|T)$
build a maximum weight spanning tree from this graph
add T as parent for each C_i
learn probabilities for $P(C_i|Parents(C_i))$
return the highest probabilistic inclusion $P(T|C') = \alpha$

Algorithm 2: Algorithm for learning probabilistic inclusions.

Description Logic Learning Results

Table: Description logic learning results

Problem	axioms, examples	DL-learner correct (length)	Combined approach correct(length)
trains	252,10	100%(5)	100%(5)
arches	47,5	100%(9)	100%(10)
moral	31,43	100%(3)	100%(5)
poker(pair)	35,49	100%(8)	100%(8)
poker (straight)	45,55	100%(5)	100%(5)

A Real World Domain

Wikipedia

Wikipedia articles consist mostly of free text, but also contain various types of structured information in the form of Wiki markup

A Real World Domain

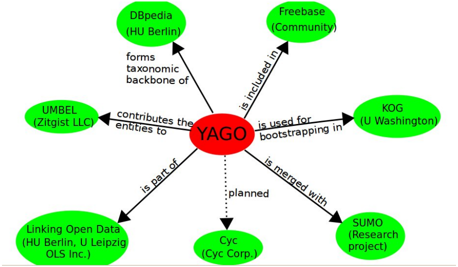
Wikipedia

Wikipedia articles consist mostly of free text, but also contain various types of structured information in the form of Wiki markup

Several Projects

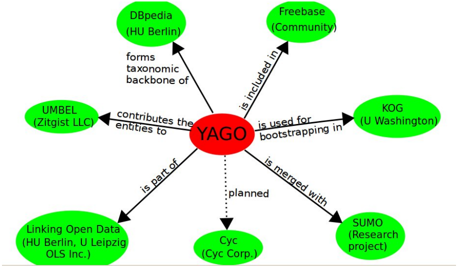
DBPedia and YAGO

YAGO



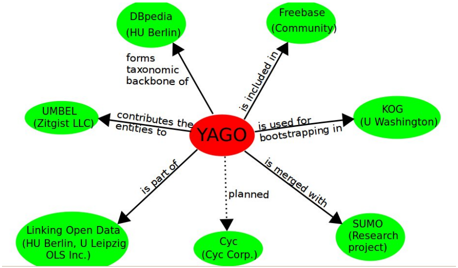
- YAGO knows more than 2 million entities (like persons, organizations, cities, etc.)

YAGO



- YAGO knows more than 2 million entities (like persons, organizations, cities, etc.)
- It knows 20 million facts about these entities: actedIn . . .

YAGO



- YAGO knows more than 2 million entities (like persons, organizations, cities, etc.)
- It knows 20 million facts about these entities: actedIn . . .
- Scientists and film directors domains

Scientists Dataset

2008 potential scientists have been considered

$$
\begin{array}{ll}
 & P(\text{wrotes}) = 0.4 \\
 & P(\text{hasAcademicAdvisor}) = 0.80 \\
 & P(\text{interestedIn}) = 0.6 \\
 & P(\text{diedOnYear}) = 0.7 \\
 & P(\text{hasWonPrize}) = 0.4 \\
 & P(\text{worksAt}) = 0.85 \\
 & P(\text{influences}) = 0.6 \\
\text{Scientist} \equiv & \text{Person} \\
 & \sqcap(\exists \text{hasAcademicAdvisor.Person} \\
 & \sqcap \exists \text{wrotes.Text} \sqcap \exists \text{worksAt.EducationalInstitution}) \\
P(\text{InfluentialScientist} & \mid \text{Scientist} \sqcap \exists \text{influences.} \\
 & \exists \text{diedOnYear.Year}) = 0.85 \\
P(\text{Musician} & \mid \text{Person} \sqcap \exists \text{hasAcademicAdvisor.} \exists \text{wrote.Text}) = 0.1 \\
\text{HonoredScientist} \equiv & \text{Scientist} \\
 & \sqcap\ \exists \text{hasWonPrize.Prize}
\end{array}
$$

Scientists II

$$\begin{aligned} P(\text{Scientist}(0) \quad & |\text{Person}(0) \\ & \sqcap(\exists \text{wrote.Text}(1) \\ & \sqcap \exists \text{hasAcademicAdvisor.Person}(2)) = 0.5 \end{aligned}$$

Scientists II

$$
\begin{aligned}
P(\text{Scientist}(0) \quad & |\text{Person}(0) \\
& \sqcap(\exists \text{wrote.Text}(1) \\
& \sqcap \exists \text{hasAcademicAdvisor.Person}(2)) = 0.5
\end{aligned}
$$

$$
\begin{aligned}
P(\text{Scientist}(0) \quad & |\text{Person}(0) \\
& \sqcap(\exists \text{wrote.Text}(1) \sqcap \exists \text{hasAcademicAdvisor.} \\
& \exists \text{influences.Person}(3))) = 0.65
\end{aligned}
$$

Directors

5589 potential directors have been considered

$$
\begin{array}{ll}
 & P(\text{isMarriedTo}) = 0.1 \\
 & P(\text{influences}) = 0.35 \\
 & P(\text{hasWonPrize}) = 0.28 \\
 & P(\text{hasChild}) = 0.05 \\
 & P(\text{diedOnYear}) = 0.5 \\
 & P(\text{directed}) = 0.8 \\
 & P(\text{actedIn}) = 0.4 \\
\text{Actor} \equiv & \text{Person} \sqcap \forall \text{actedIn.Film} \\
P(\text{Director} & \mid \text{Person} \sqcap (\exists \text{directed.Film} \sqcap \exists \text{influences.} \\
 & \exists \text{actedIn.Film}) = 0.75 \\
P(\text{FomerActor} & \mid \text{Director} \sqcap \exists \text{actedIn.Film}) = 0.6 \\
\text{HonoredDirector} \equiv & \text{Director} \sqcap \exists \text{hasWonPrize.Prize} \\
\text{FamilyDirector} \equiv & \text{Director} \sqcap (\exists \text{isMarriedTo.Director} \sqcup \exists \text{hasChild.Director}) \\
P(\text{InfluentialDirector} & \mid \text{Director} \sqcap \exists \text{hasWonPrize.Prize} \sqcap \exists \text{influences.} \\
 & \exists \text{isMarriedTo.Director}) = 0.7 \\
P(\text{MostInfluentialDirector} & \mid \text{Director} \sqcap \exists \text{diedOnYear.Year} \sqcap \exists \text{influences.} \\
 & \exists \text{hasWonPrize.Prize}) = 0.8
\end{array}
$$

Directors II

$$P(\text{Actor}(0)|\text{Person}(0) \sqcap \exists \text{actedIn.Film}(1) \sqcap \exists \text{directed.Film}(2)) = 0.4$$

$$P(\text{Director}(0)|\text{Person}(0) \sqcap \exists \text{actedIn.Film}(1) \sqcap \exists \text{directed.Film}(2)) = 0.55$$

Directors II

$$P(\text{Actor}(0)|\text{Person}(0) \sqcap \exists\text{actedIn.Film}(1) \sqcap \exists\text{directed.Film}(2)) = 0.4$$

$$P(\text{Director}(0)|\text{Person}(0) \sqcap \exists\text{actedIn.Film}(1) \sqcap \exists\text{directed.Film}(2)) = 0.55$$

$$\begin{aligned} P(\text{Actor}(0) \quad & |\text{Person}(0) \\ & \sqcap(\exists\text{actedIn.Film}(1) \sqcap \exists\text{directed.Film}(2) \\ & \sqcap\exists\text{influences.Person}(3))) = 0.3 \end{aligned}$$

Conclusions

- We have produced a combined scheme, where both the deterministic and probabilistic components receive due attention.

Conclusions

- We have produced a combined scheme, where both the deterministic and probabilistic components receive due attention.
- Initially, the search aims at finding deterministic concepts. If the score obtained is below a given threshold, a probabilistic inclusion search is conducted

Conclusions

- We have produced a combined scheme, where both the deterministic and probabilistic components receive due attention.
- Initially, the search aims at finding deterministic concepts. If the score obtained is below a given threshold, a probabilistic inclusion search is conducted
- Preliminary results have focused a real-world domain —YAGO ontology based on Wikipedia

The End

Thank you