Towards Learning Fuzzy DL Inclusion Axioms

F.A. Lisi and U. Straccia
Objective

Knowledge is inherently
1. structured
   - description in terms of objects and relations between objects
2. incomplete
   - partial description
3. vague
   - imprecise description

in many real-world domains.

We want to learn the conceptual descriptions of a target concept, given
1. the data stored into a database as fuzzy sets
2. the background knowledge about the data described via a standard
   Ontology language (specifically, OWL 2 QL)
Description Logics: the Logics behind OWL 2

- Description Logics (DLs)
  - Family of KR formalisms for incomplete structured knowledge
  - Decidable fragments of FOL
  - Expressive power depending on the set of constructors
  - Very expressive DLs at the basis of the W3C OWL 2 standard language for ontologies

- Fuzzy DLs: based on Mathematical Fuzzy Logic
  - Theoretical foundation of KR formalisms for vague knowledge
  - Truth of statements is a matter of degree (score) measured on an ordered scale ([0, 1])
  - A fuzzy interpretation $I$ maps each basic statement $p_i$ into $[0, 1]$ and is then extended inductively to all statements
  - A fuzzy set $R$ over a countable crisp set $X$ is a function $R : X \rightarrow [0, 1]$
Fuzzy DL-Lite

- DL-Lite
  - DL behind the OWL 2 QL profile
  - Especially aimed at data intensive applications
  - Tractable query answering
- Fuzziness with Gödel logic
  - \( a \otimes b = \min(a, b) \)
  - \( a \Rightarrow b = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases} \)
- Ontology-based access to a relational database

---

1(Calvanese et al., 2006)
2(Straccia, 2010)
Information is stored in a knowledge base $K = \langle F, O, A \rangle$ where:

- $F$ is a finite set of facts of the form

  $$R(c_1, \ldots, c_n)[s]$$  \hspace{1cm} (1)

- $O$ is a finite set of inclusion axioms having the form

  $$Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr$$  \hspace{1cm} (2)

  where $m \geq 1$

  - all $Rl_i$ (left-hand relation) and $Rr$ (right-hand relation) have the same arity

  $$Rl \rightarrow A \mid R[i_1, i_2]$$
  $$Rr \rightarrow A \mid R[i_1, i_2] \mid \exists R.A$$

- $A$ is a finite set of abstraction statements of the form

  $$R \leftrightarrow (c_1, \ldots, c_n)[c_{\text{score}}].sql$$  \hspace{1cm} (3)

Information can be retrieved from $K$ by means of ranking queries, i.e. conjunctive queries with a scoring function to rank the answers

$$q(x)[s] \leftarrow \exists y \ R_1(z_1)[s_1], \ldots, R_l(z_l)[s_l],$$

$$\text{OrderBy}(s = f(s_1, \ldots, s_l, p_1(z'_1), \ldots, p_h(z'_h)))$$  \hspace{1cm} (4)

where $p_i$ is a fuzzy set

Implemented in the SoftFacts system

Learning Fuzzy DL-Lite Inclusion Axioms

- the target concept $H$ is a DL-Lite atomic concept;
- the background theory $\mathcal{K}$ is a fuzzy DL-Lite knowledge base $\langle F, O, A \rangle$;
- the training set $\mathcal{E}$ is a collection of fuzzy DL-Lite like facts of the form (1) and labeled as either positive or negative examples for $H$. We assume that $F \cap \mathcal{E} = \emptyset$;
- the target theory $\mathcal{H}$ is a set of inclusion axioms of the form

$$B \sqsubseteq H$$

(5)

where $H$ is an atomic concept, $B = C_1 \sqcap \ldots \sqcap C_m$, and each concept $C_i$ has syntax

$$C \quad \rightarrow \quad A \mid \exists R.A \mid \exists R.\top.$$  

(6)
A FOIL-like algorithm

- The coverage relation for a concept $C \neq H$
  \[
  \mathcal{I}_{ILP} \models C(t) \text{ iff } K \cup \mathcal{E} \models C(t)[s] \text{ and } s > 0 .
  \]  
  (7)
- The confidence degree of an inclusion axiom is:
  \[
  cf(B \sqsubseteq H) = \frac{\sum_{t \in P} B(t) \Rightarrow H(t)}{|D|}
  \]  
  (8)

where
  - $P = \{ t \mid \mathcal{I}_{ILP} \models C_i(t) \text{ and } H(t)[s] \in \mathcal{E}^+ \}$;
  - $D = \{ t \mid \mathcal{I}_{ILP} \models C_i(t) \text{ and } H(t)[s] \in \mathcal{E} \}$;
  - $B(t) \Rightarrow H(t)$ denotes the degree to which the implication holds for the instance $t$;
  - $B(t) = \min(s_1, \ldots, s_n)$, with $K \cup \mathcal{E} \models C_i(t)[s_i]$;
  - $H(t) = s$ with $H(t)[s] \in \mathcal{E}$.
- The information gain function uses the above formulas
  \[
  \text{Gain}(cf(r'), cf(r)) = p \ast (\log_2 cf(r') - \log_2 cf(r))
  \]

where $p$ is the number of distinct positive examples covered by the inclusion axiom $r$ that are still covered by $r'$. 
function FOIL-Learn-Set-of-Axioms($H$, $E^+$, $E^-$, $K$): $H$
begin
1. $H \leftarrow \emptyset$;
2. while $E^+ \neq \emptyset$ do
3. \hspace{1em} $r \leftarrow$ FOIL-Learn-One-Axiom($H$, $E^+$, $E^-$, $K$);
4. \hspace{1em} $H \leftarrow H \cup \{r\}$;
5. \hspace{1em} $E^+_r \leftarrow \{e \in E^+ | K \cup r \models e\}$;
6. \hspace{1em} $E^+ \leftarrow E^+ \setminus E^+_r$;
7. endwhile
8. return $H$
end

Towards Learning Fuzzy DL Inclusion Axioms
Learning One Inclusion Axiom

function FOIL-Learn-One-Axiom($H$, $E^+$, $E^-$, $K$): $r$
begin
1. $B(x) \leftarrow \top$;
2. $r \leftarrow \{B(x) \rightarrow H(x)\}$;
3. $E_r^- \leftarrow E^-$;
4. while $cf(r) < \theta$ and $E_r^- \neq \emptyset$ do
5. \hspace{1em} $B_{best}(x) \leftarrow B(x)$;
6. \hspace{1em} maxgain $\leftarrow 0$;
7. \hspace{1em} foreach $l \in K$ do
8. \hspace{2em} gain $\leftarrow$ Gain($cf(B(x) \land l(x) \rightarrow H(x))$, $cf(B(x) \rightarrow H(x))$);
9. \hspace{2em} if gain $\geq$ maxgain then
10. \hspace{3em} maxgain $\leftarrow$ gain;
11. \hspace{3em} $B_{best}(x) \leftarrow B(x) \land l(x)$;
12. \hspace{2em} endif
13. endforeach
14. $r \leftarrow \{B_{best}(x) \rightarrow H(x)\}$;
15. $E_r^- \leftarrow E_r^- \setminus \{e \in E^- | K \cup r \models e\}$;
16. endwhile
17. return $r$
end
A Refinement Operator

1. Add atomic concept ($A$)
2. Add complex concept by existential role restriction ($\exists R. \top$)
3. Add complex concept by qualified existential role restriction ($\exists R.A$)
4. Replace atomic concept ($A$ replaced by $A'$ if $A' \sqsubseteq A$)
5. Replace complex concept ($\exists R.A$ replaced by $\exists R.A'$ if $A' \sqsubseteq A$)
Example: Hotel database

<table>
<thead>
<tr>
<th>HotelTable</th>
<th>RoomTable</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>rank</td>
</tr>
<tr>
<td>h1</td>
<td>3</td>
</tr>
<tr>
<td>h2</td>
<td>5</td>
</tr>
<tr>
<td>h3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tower</th>
<th>Park</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id</td>
</tr>
<tr>
<td>t1</td>
<td>p1</td>
</tr>
<tr>
<td></td>
<td>p2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DistanceTable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>from</td>
</tr>
<tr>
<td>d1</td>
<td>h1</td>
</tr>
<tr>
<td>d2</td>
<td>h2</td>
</tr>
<tr>
<td>d3</td>
<td>h3</td>
</tr>
</tbody>
</table>
Example: Abstraction statements

\[
\text{Hotel} \mapsto (h.\text{id}). \quad \text{SELECT } h.\text{id} \\
\quad \text{FROM HotelTable } h
\]

\[
\text{hasRank} \mapsto (h.\text{id}, h.\text{rank}). \quad \text{SELECT } h.\text{id}, h.\text{rank} \\
\quad \text{FROM HotelTable } h
\]

\[
\text{cheapPrice} \mapsto (h.\text{id}, r.\text{price})[\text{score}]. \quad \text{SELECT } h.\text{id}, r.\text{price}, \text{cheap}(r.\text{price}) \text{ AS score} \\
\quad \text{FROM HotelTable } h, \text{RoomTable } r \\
\quad \text{WHERE } h.\text{id} = r.\text{hotel} \\
\quad \text{ORDER BY score}
\]

\[
\text{closeTo} \mapsto (\text{from}, \text{to})[\text{score}]. \quad \text{SELECT } d.\text{from}, d.\text{to} \text{ closedistance}(d.\text{time}) \text{ AS score} \\
\quad \text{FROM DistanceTable } d \\
\quad \text{ORDER BY score}
\]

\[
\text{cheap}(p) = \text{leftshoulder}(p; 50, 100) \\
\text{closedistance}(d) = \text{leftshoulder}(d; 5, 25)
\]
Park ⊆ Attraction
Tower ⊆ Attraction
Attraction ⊆ Site
Hotel ⊆ Site
Example: Learning

- $H = \text{GoodHotel}$
- $\mathcal{E}^+ = \{ \text{GoodHotel}(h1)[0.6], \text{GoodHotel}(h2)[0.8] \}$
- $\mathcal{E}^- = \{ \text{GoodHotel}(h3)[0.4] \}$.
- $r_0 : \top \sqsubseteq \text{GoodHotel}$
- $r_1 : \text{Hotel} \sqsubseteq \text{GoodHotel}$
- $r_2 : \text{Hotel} \sqcap \exists \text{cheapPrice}. \top \sqsubseteq \text{GoodHotel}$
- $r_3 : \text{Hotel} \sqcap \exists \text{cheapPrice}. \top \sqcap \exists \text{closeTo}. \text{Attraction} \sqsubseteq \text{GoodHotel}$
- $r_4 : \text{Hotel} \sqcap \exists \text{cheapPrice}. \top \sqcap \exists \text{closeTo}. \text{Park} \sqsubseteq \text{GoodHotel}$
- $r_5 : \text{Hotel} \sqcap \exists \text{cheapPrice}. \top \sqcap \exists \text{closeTo}. \text{Tower} \sqsubseteq \text{GoodHotel}$
- Consequence:
  - $cf(r_3) = \frac{0.75 \Rightarrow 0.6 + 0.4 \Rightarrow 0.8}{3} = \frac{0.6 + 1.0}{3} = 0.5333$.
  - $cf(r_4) = \frac{0.4 \Rightarrow 0.8}{2} = \frac{0.4}{2} = 0.2$.
  - $cf(r_5) = \frac{0.8 \Rightarrow 0.6}{2} = \frac{0.6}{2} = 0.3$.
  - $\text{Gain}(r_4, r_3) = 1 \times (\log_2(0.2) - \log_2(0.5333)) = (-2.3219 + 0.907) = -1.4149$
  - $\text{Gain}(r_5, r_3) = 1 \times (\log_2(0.3) - \log_2(0.5333)) = (-1.7369 + 0.907) = -0.8299$
- $r_5$ preferred to $r_4$ as refinement of $r_3$
- $r_5$ turns out to be consistent w.r.t. $\mathcal{E}$
- $r_5$ becomes part of $\mathcal{H}$
Conclusions

- Method for inducing fuzzy DL-Lite inclusion axioms
- Extension of FOIL in a twofold direction
  - from crisp to fuzzy
  - from rules to inclusion axioms

Future work

- To implement and experiment our method
- To analyse the effect of the different implication functions and other parameters in the learning process