Inductive logic programming

Simon Jacquet

Faculty of Computer Science
Unamur

November 23rd 2021

Overview

- From previously:
  1. Equations
  2. Substitution
  3. $\bot_i$
  4. Modes
  5. Algorithms
- Algorithms:
  1. Construct C from $\bot_i$
  2. Cover set algorithm
- Difficulties:
  1. $\bot_i$
  2. Progol

UNIVERSITÉ DE NAMUR

Previously: equations

$$B \wedge H \models E \quad (1)$$
$$B \wedge \overline{E} \models \overline{H} \quad (2)$$
$$B \wedge \overline{E} \models \overline{\bot} \quad (3)$$
$$\overline{\bot} \models \overline{H} \quad (4)$$
$$H \models \bot \quad (5)$$
$$B \wedge \overline{e} \wedge \bot \vdash_h \square \quad (6)$$

- B: background knowledge
- H: hypothesis
- E: examples
- $\overline{\bot}$: set of all true literals (wrt. $B \wedge \overline{E}$)
- $\bot$: most specific clause

UNIVERSITÉ DE NAMUR

Previously: substitution

Substitution

Let θ be a substitution of the form $\{v_1/t_1, ..., v_n/t_n\}$.
Let F be an arbitrary atom.
$F\theta$ is the atom F where each of its variable v_i have been replaced by t_i.
Ex:

- $F : parent(X, Y)$
- $\theta : \{X/\text{jean},\ Y/\text{bob}\}$
- $F\theta : parent(\text{jean}, \text{bob})$

Clause substitution

The same can be done for a clause C:

- $F : parent(X, Y) :- father(X, Y).$
- $\theta : \{X/\text{jean},\ Y/\text{bob}\}$
- $F\theta : parent(\text{jean}, \text{bob}) :- father(\text{jean}, \text{bob}).$

Previously: $\bot_i$

Depth $d(v)$

$$d(v) = \begin{cases} 0, & \text{if } v \text{ is in the head of } C \\ (\min_{u \in U_v} d(u)) + 1, & \text{otherwise} \end{cases}$$

where U_v are the variables in atoms in the body of C containing v.
Ex:

- C: p(A) :- p(A,B), g(B,C), f(C,D)
- $d(A) = 1,\ d(B) = 2,\ d(C) = 3,\ d(D) = 3$

- $\bot$ can have an infinite cardinality
- $\bot_i$ is more restrained
  - the distance of its variables is $\leq i$

Previously: modes

Horn clause

$C : A \leftarrow B_1, ..., B_n$

- A is the head of clause C
- $B_1, ..., B_n$ is the body of clause C

Mode declaration

- modeh(n,atom) | modeb(n,atom)
- modeh(*,f(+int,-int)), modeb(*,d(+int,-int))
- Use *modeh* for the head of a clause, *modeb* for its body

UNIVERSITÉ DE NAMUR

Previously: modes (ii)

Instantiation

- Let M be a set of modes
- $m \in M$ is a mode declaration
- $a(m)$ is the atom of m with place markers replaced

Ex

$$m = \text{modeh}(*, f(+\text{int}, -\text{int}))$$
$$a(m) = f(A, B)$$

Previously: algorithms

- Three algorithms:
  - Cover set algorithm
  - Construct $\perp_i$
  - Construct C form $\perp_i$

Algorithms: Cover set algorithm

Algorithm 1: Cover set algorithm

input: h, i, B, M, E

forall $e \in E$ **do**

- Construct $\bot_i$ for e using Algorithms 3 and 4
- Construct state s from $\bot_i$ using Algorithm 2
- Let C' be the unflattening of $C(s)$
- $B \leftarrow B \cup C'$
- $E \leftarrow E - \{e : e \in E,\ B \wedge \overline{e} \vdash_h \emptyset\}$

Algorithms: Construct C

Algorithm 2: Construct C

input: $h, B, e, \perp_i$

1 Open $\leftarrow \{\langle \square, \emptyset, 1 \rangle\}$, Closed $\leftarrow \emptyset$
2 **while** *True* **do**
3 $s \leftarrow$ best(Open)
4 Open $\leftarrow$ Open $-\{s\}$, Closed $\leftarrow$ Closed $\cup\{s\}$
5 **if** $\neg$ *prune(s)* **then**
6 Open $\leftarrow$ (Open $\cup\, \rho(s)$) $-$ Closed
7 **if** *terminated(Closed,Open)* **then**
8 **return** best(Closed)
9 **if** *Open* $= \emptyset$ **then**
10 print 'no compression'
11 **return** $\langle e, \emptyset, 1 \rangle$

- ρ is the refinement operator

Algorithms: Constructing $\perp_i$ (i)

Algorithm 3: Construct $\perp_i$ - Part 1

Get $m \in M$, modeh such that $a(m) \preceq a$ with substitution θ_h
if $\nexists m$ **then**
    return $\square$
$a_h \leftarrow a(m)$
for $v/t \in \theta_h$ **do**
    if *v corresponds to #type in m* **then**
        Replace v by t in a_h
    else
        Replace v by v_k in a_h, with $k = \text{hash}(t)$
    if *v corresponds to +type in m* **then**
        Add v to InTerms
Add a_h to $\perp_i$

UNIVERSITÉ DE NAMUR

Algorithms: Constructing $\perp_i$ (ii)

Algorithm 4: Construct $\perp_i$ - Part 2

```
1  for k ← 1, ..., i do
2      forall modeb m ∈ M do
3          Let {v_1, ..., v_n} be the variables corresponding to +type in a(m)
4          Let T_i be the set of all terms of the type associated with v_i in m
5          T(m) ← T_1 × ... × T_n
6          forall ⟨t_1, ..., t_n⟩ ∈ T(m) do
7              a_b ← a(m)
8              θ ← {v_1/t_1, ..., v_n/t_n}
9              if Prolog succeeds on goal a_bθ then
10                 Let Θ_b be the set of answer substitutions
11                 forall θ_b ∈ Θ_b do
12                     forall v/t ∈ θ_b do
13                         if v corresponds to #type in m then
14                             Replace v by t in a_b
15                         else
16                             Replace v by v_k in a_b, with k = hash(t)
17                         if v corresponds to -type then
18                             Add v to InTerms
19                 Add ā_b to ⊥_i
20 return ⊥_i
```

Important numbers

```
f = Number of positive examples covered -
    Number of negative examples covered -
    Number of literals in body of clause -
    Optimistic estimate of literals needed
p = Number of positive examples covered
n = Number of negative examples covered
h = Optimistic estimate of literals needed
```

UNIVERSITÉ DE NAMUR

Example of execution

```
[:- modeb(1,dec(+int,-int))? - Time taken 0.00s]
[:- modeb(1,plus(+int,+int,-int))? - Time taken 0.00s]
[:- modeb(1,inc(+int,-int))? - Time taken 0.00s]
[:- modeb(1,mult(+int,+int,-int))? - Time taken 0.00s]
[:- modeb(1,plus(+int,+int,-int))? - Time taken 0.02s]
[:- modeh(1,plus(+int,+int,-int))? - Time taken 0.00s]
[:- modeh(1,mult(+int,+int,-int))? - Time taken 0.00s]
[:- determination(plus/3,dec/2), determination(plus/3,inc/2), determination(plus/3,plus/3)? - Time taken 0.02s]
[:- determination(mult/3,dec/2), determination(mult/3,plus/3), determination(mult/3,mult/3)? - Time taken 0.00s]
[:- commutative(mult/3), commutative(plus/3)? - Time taken 0.00s]
[:- set(c,3)? - Time taken 0.00s]
[Testing for contradictions]
[No contradictions found]
[Generalising plus(4,X,Y) :- inc(X,U), inc(U,V), inc(V,W), inc(W,Y).]
[Most specific clause is]

plus(A,B,C) :- dec(A,D), inc(B,E), dec(D,F), plus(D,B,G), plus(D,
        E,C), inc(D,A), inc(E,H), dec(F,I), plus(F,B,H), plus(F,
        E,G), plus(F,F,A), plus(F,H,C), inc(F,D), inc(G,C), inc(H,
        G).

[C:-2,6,5,3 plus(A,B,C).]
[C:-3,4,4,2 plus(A,B,C) :- dec(A,D).]
[C:-3,4,4,1 plus(A,B,C) :- dec(A,D), inc(B,E).]
[C:1,4,0,0 plus(A,B,C) :- dec(A,D), inc(B,E), plus(D,E,C).]
[C:-3,5,5,2 plus(A,B,C) :- inc(B,D).]
[5 explored search nodes]
[f=1,p=4,n=0,h=0]
[Result of search is]

plus(A,B,C) :- dec(A,D), inc(B,E), plus(D,E,C).
```

UNIVERSITÉ DE NAMUR

Difficulties